目 录

第1章　MATLAB概论

1.1　MATLAB简介

MATLAB 是功能强大的科学及工程计算软件，它不但具有以矩阵计算为基础的强大数学计算和分析功能，而且还具有丰富的可视化图形表现功能和方便的程序设计能力。MATLAB 的应用领域极为广泛，除数学计算和分析外，还被广泛地应用于自动控制、系统仿真、数字信号处理、图形图像分析、数理统计、人工智能、虚拟现实技术、通信工程、金融系统等领域，因此，MATLAB 是面向 21 世纪的计算机程序设计及科学计算语言。

1.1.1　MATLAB 6 的新特点

MATLAB 6.1 是 MathWorks 公司 2001 年 6 月推出的最新版本。与以前版本相比较，MATLAB 6 作了很多根本性的改进，其主要特点为：

(1) 改变了 5.X 以前版本的单一命令窗口方式，推出了一个全新的开发环境，它将一些常用的交互式工作界面高度地集成于操作桌面，使工作环境发生了质的变化。

(2) 加强了基本平台函数和工具箱函数的可视化功能，许多常用的功能和函数都由单纯的命令行方式改为交互式图形界面方式，使用更加方便。

(3) 在数值处理方面增加了许多新的功能，不仅增加了许多新函数，还更新了部分函数的功能和算法。

(4) 在程序设计方面向面向对象的编程方式过渡，引入了类和对象的概念，引入了函数句柄的概念，用户可以为任何 MATLAB 函数创建函数句柄，为大型程序设计提供了方便。

(5) 在命令窗口增加了错误跟踪功能，当运行 M 文件出现错误时，点击命令窗口中带有下划线的错误信息，就可以打开出现错误的 M 文件，并找到相应的程序行。

(6) 在一些常用的窗口中增加了一些实用的功能。如在绘图窗口中增加了曲线拟合、数据统计等交互工具，在工作空间窗口中增加了各种绘图工具等。

(7) 对编辑/调试器的界面及功能做了改进，增加了行号和书签等功能。

(8) 提供了新的图形对象属性编辑器，功能更加强大，使用更加方便。

(9) 提供了一个与以前完全不同的图形用户界面开发工具，将 MATLAB 带入了一个以图形用户界面设计为基础的程序开发的新阶段，在使用上更加方便灵活。

(10) 对部分工具箱的功能进行了改进和加强，特别是 MATLAB 6.1 新增加了虚拟现实工具箱，采用标准的虚拟现实建模语言（VRML）技术，可以建立由 MATLAB 和 Simulink 环境驱动的动态三维场景。

(11) 增加了与 Java 的接口，并为二者的数据交换提供了相应的程序库。

1.1.2 MATLAB 的主要组成部分

MATLAB 系统包括 5 个主要部分：

(1) 开发环境

MATLAB 开发环境由一组工具和组件组成，这些工具是图形化的用户界面，包括 MATLAB 桌面和命令窗口、命令历史窗口、帮助信息浏览器、工作空间浏览器、文件和搜索路径浏览器。

(2) MATLAB 数学函数库

MATLAB 集成了丰富的数学函数库，其强大的计算能力覆盖了从基本函数（如求和、正弦、余弦和复数运算等）到高级函数（如矩阵求逆、矩阵特征值、贝塞尔函数和快速傅里叶变换等）的范围。

(3) MATLAB 语言

MATLAB 语言是一种以矩阵运算为基础的高级语言，包括控制流的描述、函数、数据结构、输入输出及面向对象的编程环境，既可以编制快速使用小程序，也可以编制大型复杂的应用程序。

(4) 图形功能

MATLAB 提供了功能强大的图形系统，既可以用高级命令完成二维和三维数据的可视化、图像处理、动画和图形表达等功能，也可以通过使用图形句柄完成复杂的图形功能，实现对所有图形对象的操作。

(5) 应用程序接口（API）

MATLAB 还提供了应用程序接口库函数，允许用户使用 C 或 FORTRAN 语言编写程序与 MATLAB 连接，功能包括与 MATLAB 的动态连接、调用 MATLAB 作为运算引擎、读写 MAT 文件等。

MATLAB 的广泛应用促进了其本身功能的迅速发展，以特定领域的应用为主要目的的应用程序——工具箱的数量和内容在不断增多。截止到 MATLAB 6.1，仅由 MathWorks 公司发售的 MATLAB 工具箱就已达到 20 余个，其他第三方工具箱更多达数百个之多。

1.2 MATLAB 的安装

MATLAB 6 以后的版本由于采用了新的图形系统，对计算机的要求较以前的版本有了较大的提高。为了保证 MATLAB 的可靠运行，系统的配置至少应达到：

CPU　Pentium II 以上

显示卡　支持真彩色，分辨率在 800×600 点以上，最好达到 1 024×768

内存　64 MB 以上

MATLAB 可以在 Windows 98/Me/2000/XP 等操作系统上安装，同时还提供了 UNIX 等平台下的版本。但由于 Windows NT 操作系统推出的较早，在 Windows NT 下安装 MATLAB 有时会出现问题，建议尽量不要在 NT 下安装。

在 Windows 98/Me/2000/XP 下安装 MATLAB 6.1 的步骤如下：

(1) 在光驱中放入 MATLAB 安装盘，系统自动启动安装程序，显示“欢迎安装 MATLAB”对话框（图 1-1），声明版权信息。单击“Next”继续。

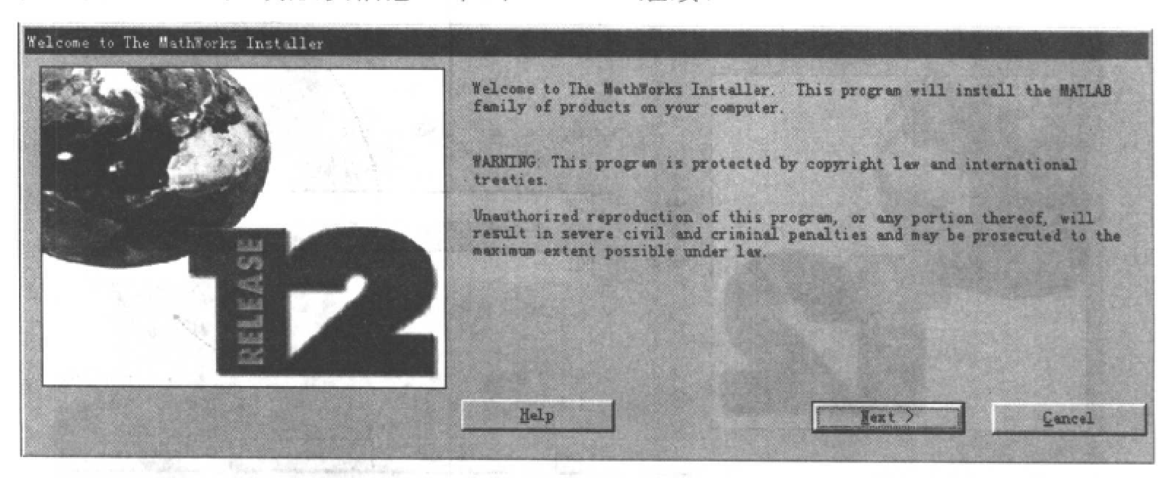


图 1-1 “欢迎安装 MATLAB”对话框

(2) 显示“个人许可密码”对话框（图 1-2），要求输入个人许可密码（PLP）。输入密码后，单击“Next”继续。

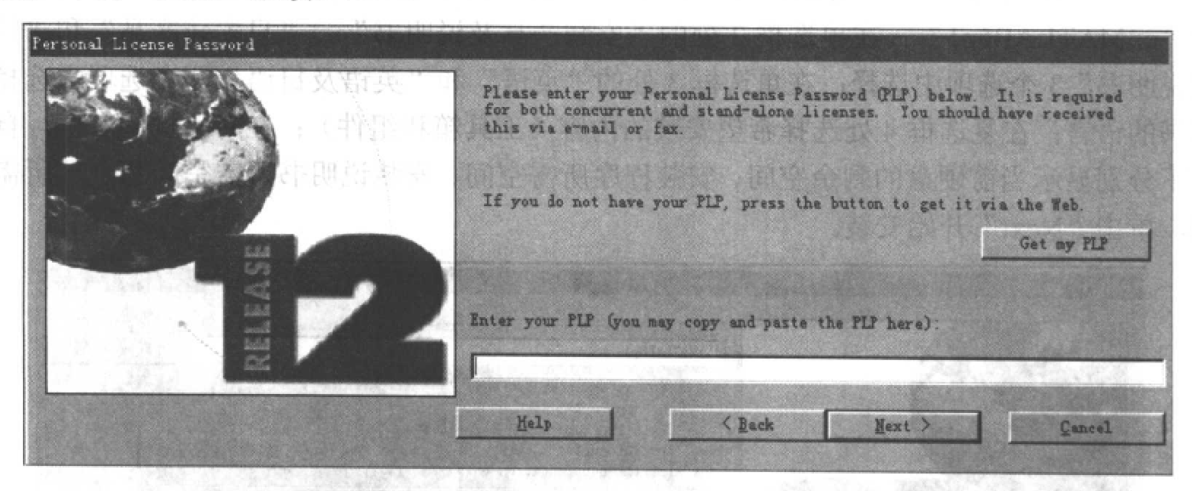


图 1-2 “个人许可密码”对话框

(3) 显示“许可协议”对话框（图 1-3），单击“Yes”继续。

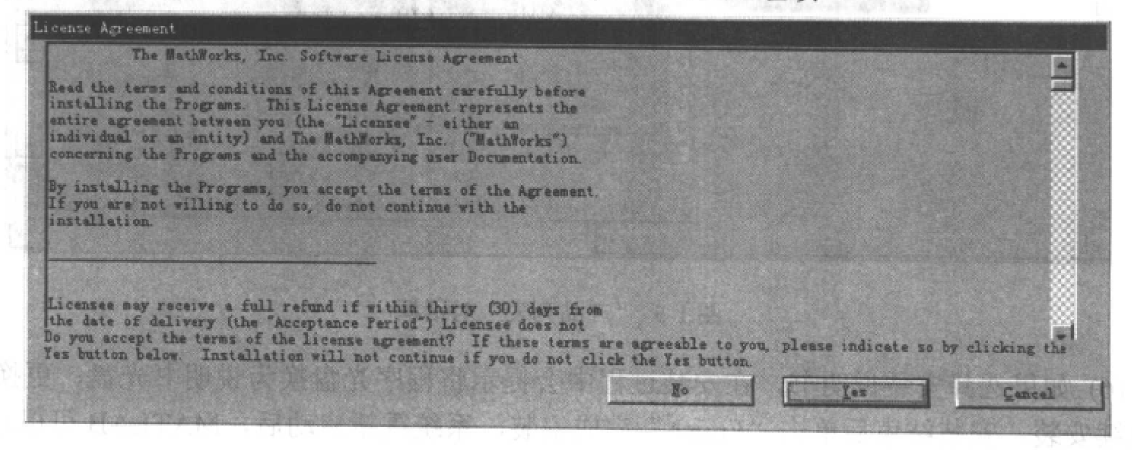


图 1-3 “许可协议”对话框

(4) 显示“用户信息”对话框（图1-4），按要求输入用户姓名和单位，单击“Next”继续。

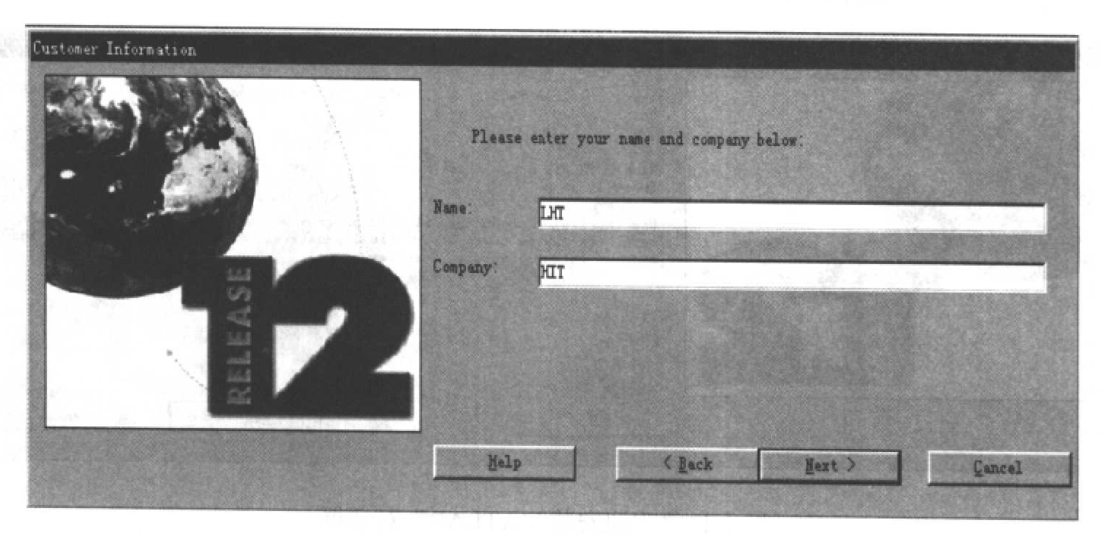


图1-4 “用户信息”对话框

(5) 显示“产品列表”对话框（图1-5），在文本框1处选择安装路径，缺省的安装路径为“c:\MATLAB6p1”；在单选框2处的“安装产品及说明书”、“只安装产品”和“只安装说明书”3个选项中选择；在单选框3处的“英语”和“英语及日语”两个选项中选择说明书的语言；在复选框4处选择希望安装的内容（工具箱和组件）；在对话框的右侧，自上而下分别显示当前硬盘的剩余空间、安装程序所需空间、安装说明书所需空间和全部所需空间，单击“Next”开始安装。

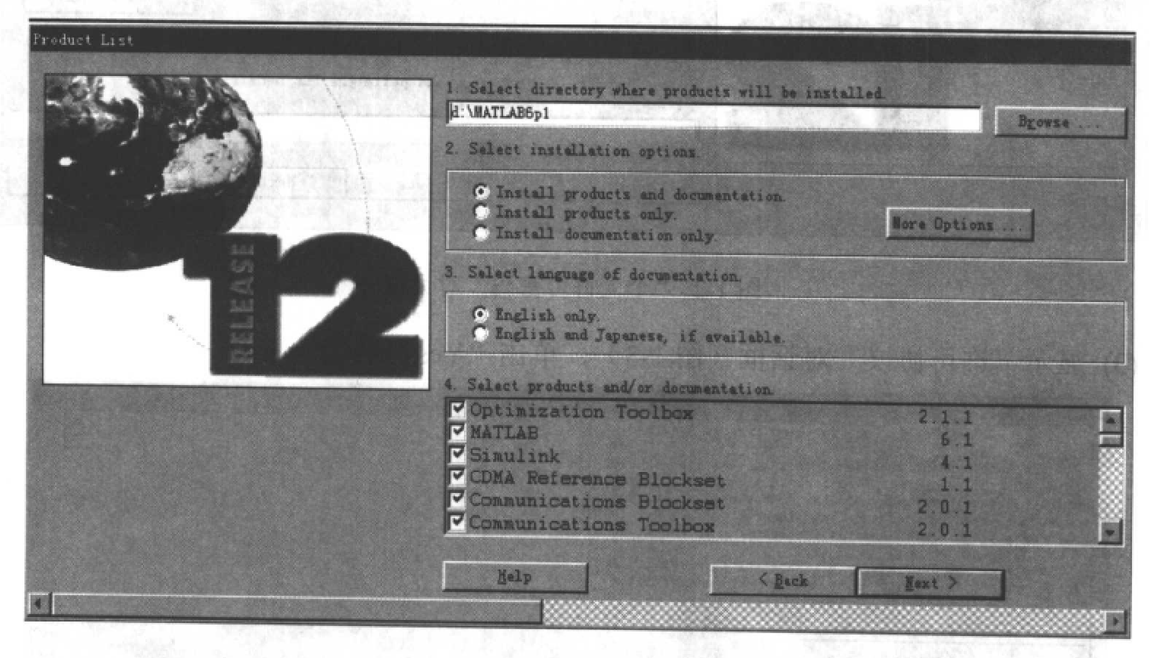


图1-5 “产品列表”对话框

(6) 如果选择安装说明书，在安装过程中会提示将程序光盘换为说明书光盘，更换光盘后继续安装，安装结束后单击“Finish”完成安装，系统重新启动后，MATLAB可以运行。

1.3 MATLAB 开发环境

MATLAB 6 除保留了传统的命令窗口外，还增加了启动平台窗口、工作空间窗口、命令历史窗口、当前路径窗口等窗口，与新的 M 文件编辑器和新的在线帮助浏览器等共同构成了 MATLAB 6 的开发环境。

MATLAB 的开发环境是 MATLAB 语言的基础和核心部分，MATLAB 语言的全部功能都是在 MATLAB 的开发环境中实现的，MATLAB 的仿真工具 Simulink、MATLAB 的工具箱等其他附加功能的实现也必须使用 MATLAB 开发环境，因此，掌握 MATLAB 的开发环境是掌握 MATLAB 语言的关键。

启动 MATLAB 后，将显示包括命令窗口、启动平台窗口、工作空间窗口、命令历史窗口和当前路径窗口等 5 个窗口和主菜单组成的操作桌面（图 1-6）。本节对 MATLAB 操作桌面的主菜单和各个窗口作简要介绍，部分窗口的功能和使用将在以后的章节中详细介绍。

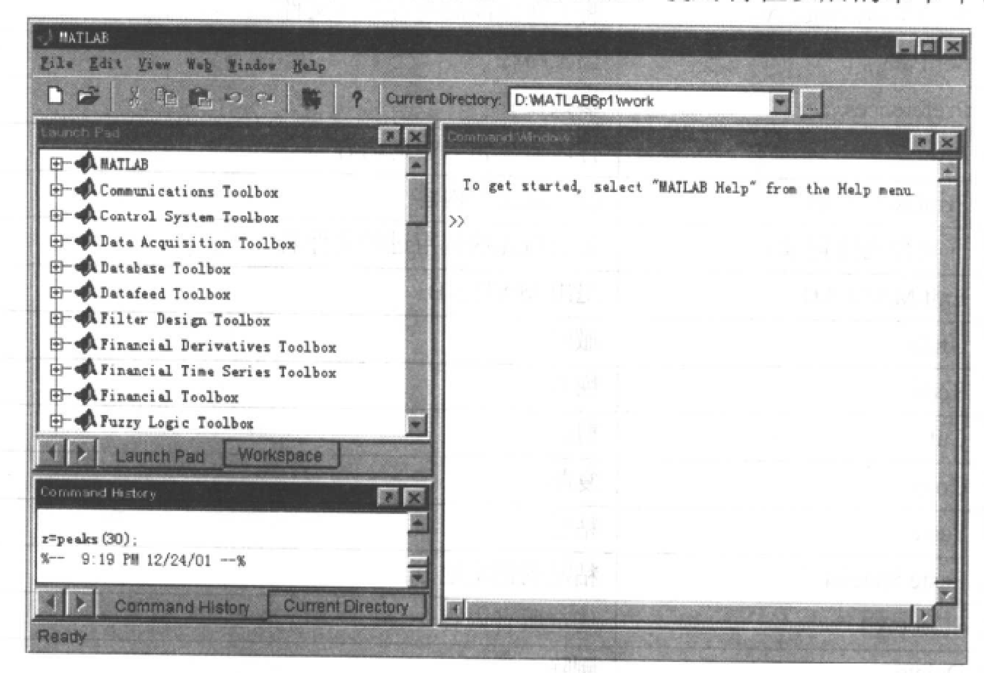


图 1-6 操作桌面

操作桌面缺省状态下显示 3 个窗口，启动平台和工作空间窗口在同一位置显示，可以通过该窗口下方的箭头或窗口标签来切换，命令历史窗口和当前目录窗口在同一位置显示，可以通过该窗口下方的箭头或窗口标签来切换。每个窗口的右上角都有两个按钮，当选择按钮时，可以使该窗口脱离操作桌面独立出来；当选择按钮时，可以关闭该窗口。也可以通过 View 菜单来选择显示哪些窗口。

MATLAB 还设定了几种特定的窗口布局方式，在 View 菜单的 Desktop Layout 选项中，给定了 6 种布局方式：

- Default（缺省方式）
- Command Window Only（只显示命令窗口方式）
- Simple（简单方式，只有命令历史窗口和命令窗口）

- Short History（低命令历史窗口方式）
- Tall History（高命令历史窗口方式）
- Five Panel（5 个窗口平铺方式）

1.3.1　主菜单和工具栏

操作桌面上包括了 MATLAB 的主菜单和带有快捷按钮的工具栏，菜单选项见表 1-1。

表 1-1　操作桌面的菜单选项

菜单项	选项	内容
File	New	建立新文件，包括 M 文件、图形文件、Simulink 模型和 GUI
	Open…	打开已存在的文件
	Close Command Window	关闭命令窗口
	Import Data ...	用输入向导（Import Wizard）导入数据
	Save Workspace As...	将工作空间的内容存入文件
	Set Path...	设置路径
	Preferences...	参数设置
	Print…	打印命令窗口中的内容
	Print Selection...	打印选定的内容
	（文件调用记录）	记录前几次调用过的文件名
	Exit MATLAB	退出 MATLAB
Edit	Undo	撤消
	Redo	恢复
	Cut	剪切
	Copy	复制
	Paste	粘贴
	Paste Special	粘贴到指定地方
	Select All	选定所有内容
	Delete	删除
	Clear Command Window	清除命令窗口
	Clear Command History	清除命令历史
	Clear Workspace	清除工作空间的内容
View	Desktop Layout	桌面布局
	Undock Command Window	脱离命令窗口（可任意选定窗口）
	Command Window	打开命令窗口
	Command History	打开命令历史窗口
	Current Directory	打开当前目录窗口
	Workspace Browser	打开工作空间浏览器
	Launch Pad	打开启动平台
	Help	打开在线帮助浏览器

续表

View	Current Directory Filter	当前目录过滤器
	Workspace View Options	工作空间观察选项
Web	The MathWorks Web Site	MathWorks 网站
	Technical Support	技术支持
	Products	产品
	Membership	成员
Help	Full Product Family Help	全部产品的帮助窗口
	MATLAB Help	MATLAB 帮助
	Using the Desktop	使用操作桌面
	Using the Workspace Browser	使用工作空间浏览器
	Demo	演示
	About MATLAB	关于 MATLAB

工具栏中各快捷按钮的功能如下：

——建立新的 M 文件、图形、Simulink 模型或 GUI 对象

——弹出打开文件对话框，打开文件

——剪切，剪切选中的内容并放入剪贴板

——复制，将选中的内容复制到剪贴板中

——粘贴，将剪贴板中的内容粘贴到指定位置

——撤消，撤消上一步操作

——恢复，恢复上一次操作

——仿真库浏览器（未安装 Simulink 时不显示该按钮）

——打开帮助浏览器

1.3.2 MATLAB 的通用参数设置

MATLAB 的通用参数和各功能窗口的参数可以通过主菜单中的 Preferences 项设置，这里先介绍通用参数的设置。

在主菜单中选择 Preferences 项，打开"Preferences"窗口（图 1-7），缺省状态为通用参数设置，其选项包括：

- Display（显示选择） "Show tooltips"（显示相关信息），当单选框选择后，鼠标放在工具栏的快捷按钮上时显示相关信息。
- Toolbox caching（工具箱缓冲区） 对于远程使用 MATLAB 的用户，应选择单选框 Enable Toolbox caching（打开工具箱缓冲区），建立一个高速缓冲区，以提高使用速度，对单机用户该选项作用不大。
- Figure window printing（图形窗口打印） 有 3 个选项，分别为"Use printer default"（按打印机缺省设置输出）、"Always send as black and white"（按黑白图形输出）和"Always

send as color”（按彩色图形输出）。

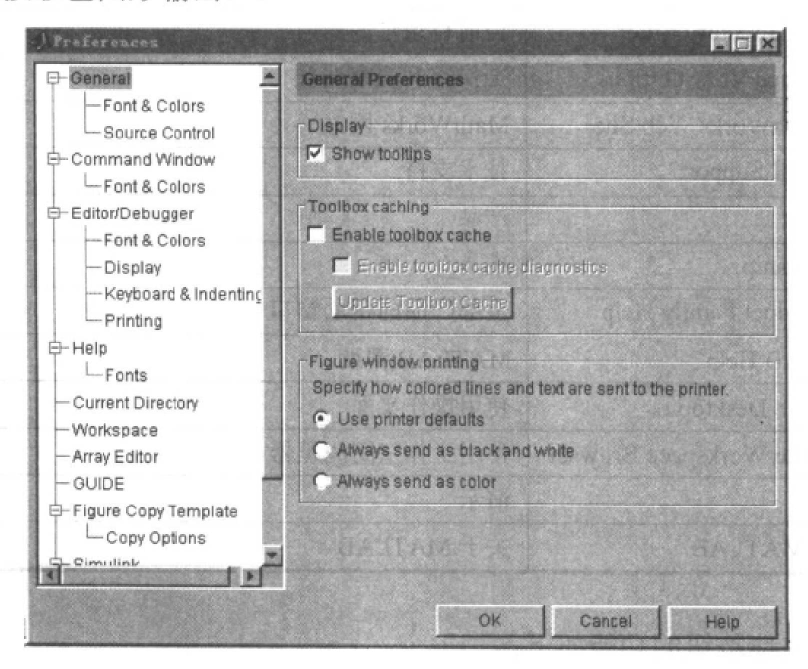


图 1-7　通用参数设置窗口

在窗口左侧的树状结构中展开 General 项，有两个可以选择的参数设置窗口：

(1) Font & Colors（字体和颜色）（图 1-8）

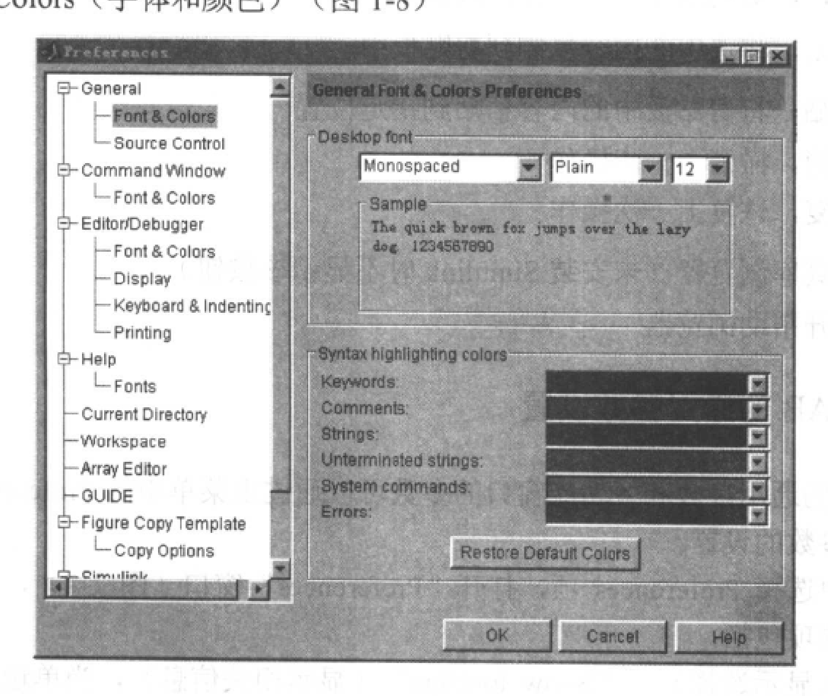


图 1-8　通用参数-字体和颜色设置窗口

• Desktop font（操作桌面字体）　该项将改变 MATLAB 操作桌面的所有窗口的字体设置（包括 M 文件编辑器、数组编辑器等），可以设置字体的名称、形式和大小。

• Syntax highlighting colors（标示符颜色）　可以选择的标示符包括 Keywords（关键词）、Comment（注释）、String（字符串）、“Unterminated string”（未完成字符串）、“System

commands”（系统命令）和 Errors（错误信息），如希望恢复系统的缺省颜色，可以按“Restore Default Colors”（恢复缺省的颜色）按钮。

(2) Source Control（控制源）（图 1-9）

可以从下拉列表中选择不同的控制源。

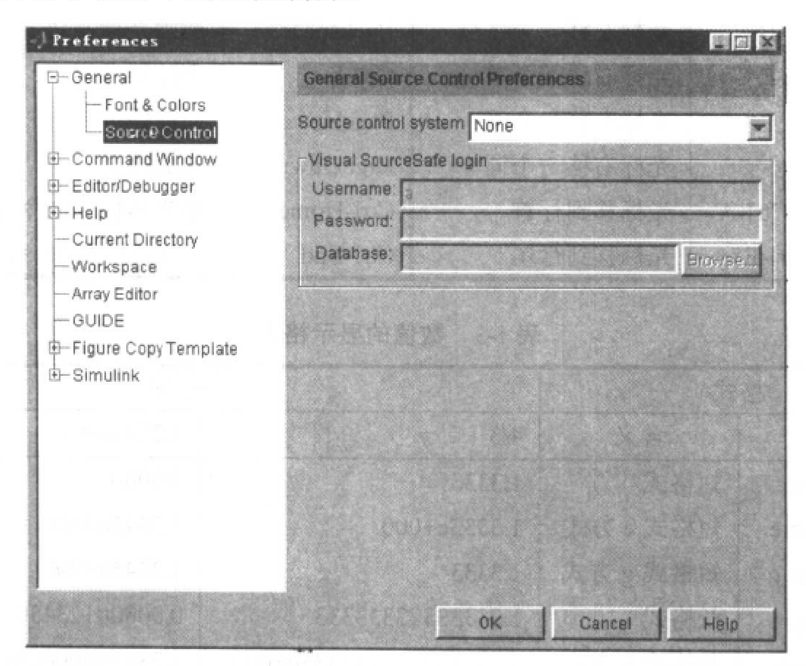


图 1-9 通用参数-控制源设置窗口

1.3.3 命令窗口

命令窗口保留了 MATLAB 传统的交互式操作功能，即在命令窗口中直接输入命令或 MATLAB 函数，系统自动反馈信息或结果。在命令窗口中，“>>”为提示符。

【例 1-1】 通过命令窗口直接输入，计算 $18+\dfrac{5\sin\frac{\pi}{6}}{2+\cos\frac{\pi}{6}}$。

在命令窗口中，只需输入：

```
>>18+(5*sin(pi/6))/(2+cos(pi/6))
```

按 Enter 键，就可以得到计算结果：

```
ans =

       18.8723
```

如果将上式改为 $18+\dfrac{5\sin\frac{\pi}{3}}{2+\cos\frac{\pi}{3}}$，不必逐一重新输入，而只需按一次↑键，调回已经输入的 18+(5*sin(pi/6))/(2+cos(pi/6))，将其中的 6 均改为 3 即可。

除↑外，还有一些命令行功能键，见表 1-2。

使用 clc 命令清除命令窗口的显示内容，但不清除工作空间，用↑键仍可调回上次输入的命令。

使用 format 命令可以控制命令窗口数值显示的格式（显示格式也可以在 Preferences 中设置，详见 1.3.4），表 1-3 为数值显示的格式。

表 1-2　常用的命令行功能键

按键	功能	按键	功能
↑，Ctrl-P	调出前一命令行	Esc	清除命令行
↓，Ctrl-N	调出后一命令行	Del，Ctrl-D	删除光标处字符
←，Ctrl-B	光标左移一个字符	Backspace	删除光标左边字符
→，Ctrl-F	光标右移一个字符	Ctrl-K	删除至行尾
Ctrl-←	光标左移一个词	PageUp	向前翻页
Ctrl-→	光标右移一个词	PageDown	向后翻页
Home，Ctrl-A	光标移到行首	Ctrl-Home	把光标移到命令窗口首
End，Ctrl-E	光标移到行尾	Ctrl-End	把光标移到命令窗口尾

表 1-3　数值的显示格式

格式		对应结果	
命令	含义	4/3	1.2345e-6
format short	短格式	1.3333	0.0000
format short e	短格式 e 方式	1.3333e+000	1.2345e-006
format short g	短格式 g 方式	1.3333	1.2345e-006
format long	长格式	1.33333333333333	0.00000123450000
format long e	长格式 e 方式	1.333333333333333e+000	1.234500000000000e-006
format long g	长格式 g 方式	1.33333333333333	1.234500000000000e-006
format hex	16 进制格式	3ff5555555555555	3eb4b6231abfd271
format +	+格式	+	+
format rat	分数格式	4/3	1/810045
format bank	银行格式	1.33	0.00

1.3.4　命令窗口的参数设置

命令窗口的参数设置如图 1-10 所示。

• Text display（文本显示）　命令窗口中文本显示格式的设置。其中“Numeric format”（数据格式）设置命令窗口数据的显示格式（见表 1-3）；“Numeric display”（数据显示方式）可以在 loose 和 compact 之间选择，选择 loose，命令窗口中的命令和结果都隔行显示，而选择 compact 则以紧凑方式显示；“Space per tab”（Tab 键控制）设置每次按 Tab 键所对应的空格数。

• Display（显示参数）　显示参数设置，其中“Echo on”（M 文件命令显示控制）控制 M 文件的运行过程中，M 文件的内部命令是否在命令窗口中显示；“Wrap line”（自动换行）：如果选择该选项，输入输出行根据命令窗口的宽度自动换行，否则，无论输入输出行的长度是多少都显示在一行，宽度不够时出现滚动条；“Limit matrix display width to eighty columns”（80 列显示限制）控制矩阵的显示列数是否为 80；“Enable up to *n* tab completions”（模糊查询的最大限制）。如果勾选该项，Tab 键为模糊查询键，输入框中的数值为模糊查询（详见 1.5.4）功能中显示结果的最大数量。例如，输入的数值为 20，如果模糊查询的结果少于

20 条，在命令窗口中显示查询结果；如果查询结果多于 20 条，则显示“There are over 20 completions”，此时可以加大数值或输入更准确的查询条件，如果不勾选该项，恢复 Tab 键的原有功能；“Command session scroll buffer size”（回调命令缓冲区）控制命令窗口中回调命令缓冲区的大小。

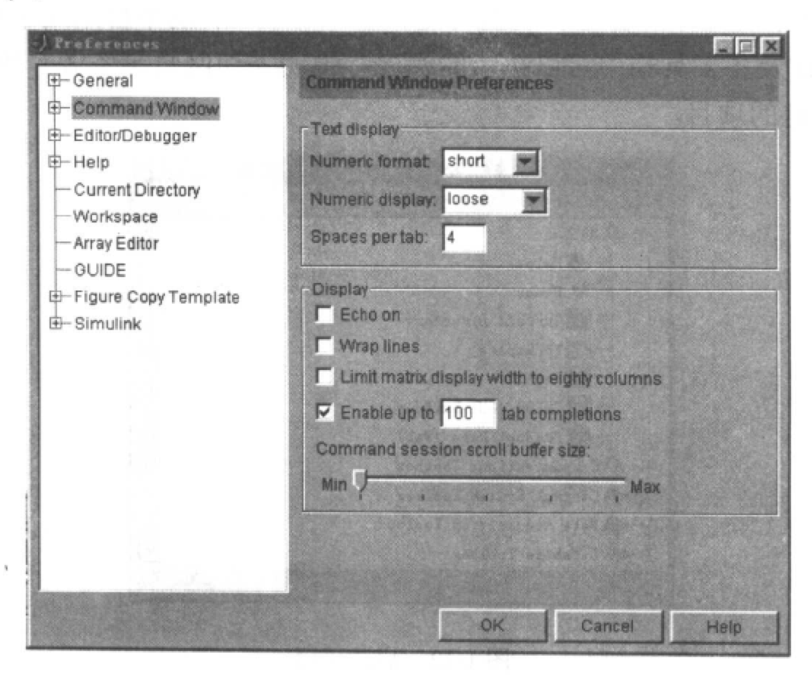


图 1-10 命令窗口设置窗口

在树状结构中展开“Command Window”项，可以设置“Font & Colors”（字体和颜色）（图 1-11）。缺省状态为使用操作桌面的字体和颜色，也可以单独设置命令窗口的字体和颜色。

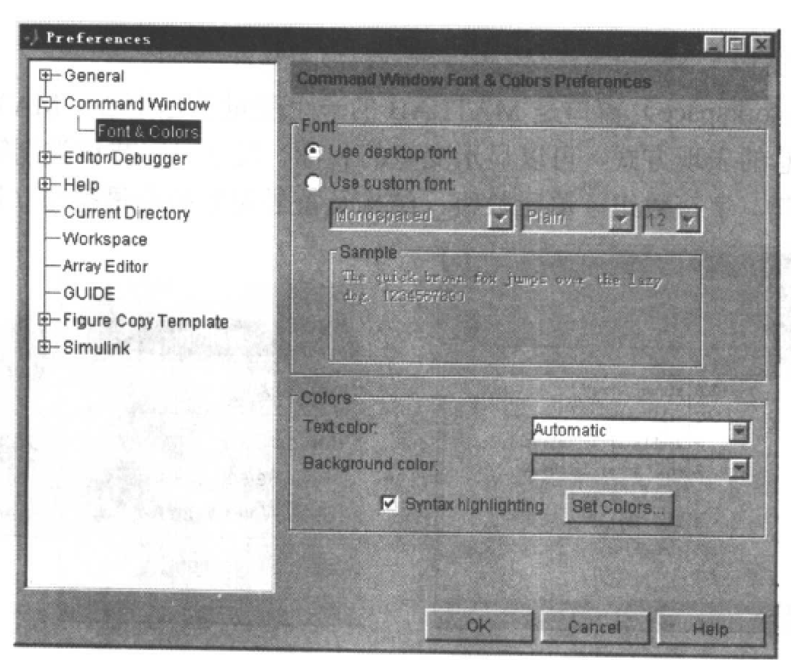


图 1-11 命令窗口-字体和颜色设置窗口

1.3.5　启动平台

在 MATLAB 6 新增加的启动平台（Launch Pad）窗口中包括了系统中已安装了的全部 MATLAB 程序和帮助文件的管理和启动功能（图 1-12）。启动平台采用了树型菜单的形式，列出了 MATLAB 及各工具箱中可以执行的程序、说明书和网页等资源，用户可以直接从启动平台进入所选定的项目。

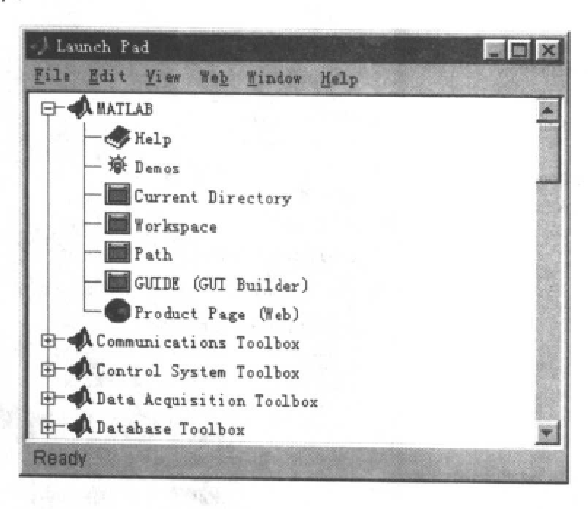


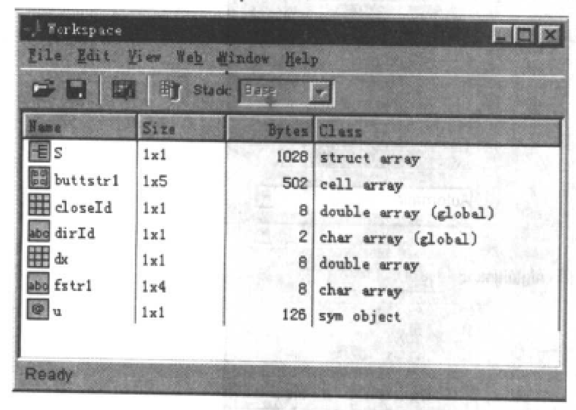
图 1-12　启动平台

启动平台实际上是由文本写成的 M 文件，扩展名为.xml，包括了标题、调用程序、图标等信息，用户也可以自行编辑（如将标题改为中文等）。

1.3.6　工作空间

1. 工作空间窗口

工作空间（Workspace）窗口是 MATLAB 的一个变量管理中心。MATLAB 6 改进了工作空间中变量信息的表现方式，可以显示变量的名称、尺寸、字节和类别等信息，同时用不同的图标表示矩阵、字符数组、单元数组、结构及符号对象等变量类型（图 1-13）。



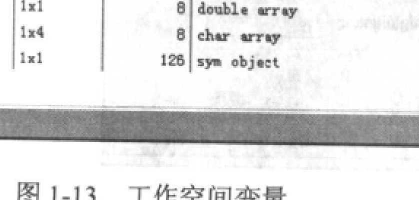


图 1-13　工作空间变量

图 1-14　工作空间窗口的弹出菜单

除了表现变量的信息外，工作空间窗口还提供了丰富的处理功能。在某一变量上单击鼠

标右键会弹出菜单（图 1-14），选择“Open Selection…”（打开）选项打开数组编辑器（图 1-15），使用数组编辑器可以对一些类型变量的格式、尺寸及元素进行修改；选择“Graph Selection”（图形选择）选项，可以使用不同的方法绘制当前变量的图形（关于 MATLAB 的绘图功能详见本书第 5 章和第 6 章）。菜单的其他选项见表 1-4。

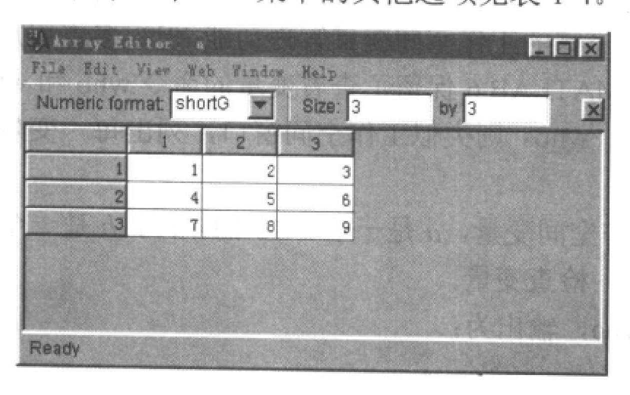


图 1-15 数组编辑器

表 1-4 工作空间窗口弹出菜单的其他选项

菜单项	功能
Select All	选择全部变量
Import Data…	导入数据（同主菜单）
Save Selection As…	将所选变量存入.mat 文件
Save Workspace As…	将全部变量存入.mat 文件
Delete Selection	删除所选变量（将弹出对话框确认是否删除）
Clear Workspace	清除工作空间（将弹出对话框确认是否清除）

2. 工作空间参数设置

工作空间的参数设置如图 1-16 所示。

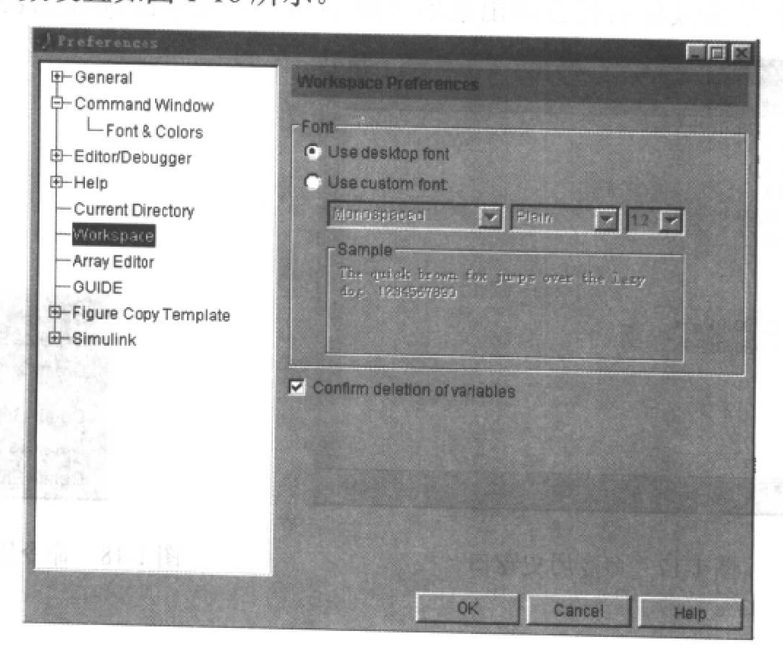


图 1-16 工作空间设置窗口

窗口的上部为字体和颜色设置，缺省状态为使用操作桌面的字体和颜色，也可以单独设置工作空间窗口的字体和颜色。下部为“Confirm deletion of variables”（确认删除变量），勾选该项则在删除工作空间变量时会弹出一个对话框，要求确认是否删除。

3. 工作空间管理的命令方式

MATLAB 6 提供了丰富的工作空间变量管理功能，但很多老用户可能更习惯于以前版本的命令行方式，因此，MATLAB 6 保留了查看变量的命令 who 和 whos。who 命令在命令窗口列出变量的简明列表，whos 则类似工作空间窗口，列出每一变量的大小、数据类型等信息。

【例 1-2】 有 3 个工作空间变量，*a* 是一个 3×3 的矩阵，*b* 是一个字符串，*c* 是一个常数，通过命令窗口直接输入，检查变量。

在命令窗口输入 who，输出为：

```
>>who
Your variables are:
a  b  c
```

在命令窗口输入 whos，输出为：

```
>>whos
Name        Size           Bytes     Class
  a          3x3             72     double array
  b          1x4              8     char array
  c          1x1              8     double array
Grand total is 14 elements using 88 bytes
```

1.3.7　命令历史窗口

MATLAB 6 的命令历史（Command History）窗口显示已执行过的命令(图 1-17)，在该窗口的某一命令上单击鼠标右键会弹出菜单（图 1-18），菜单的选项见表 1-5。

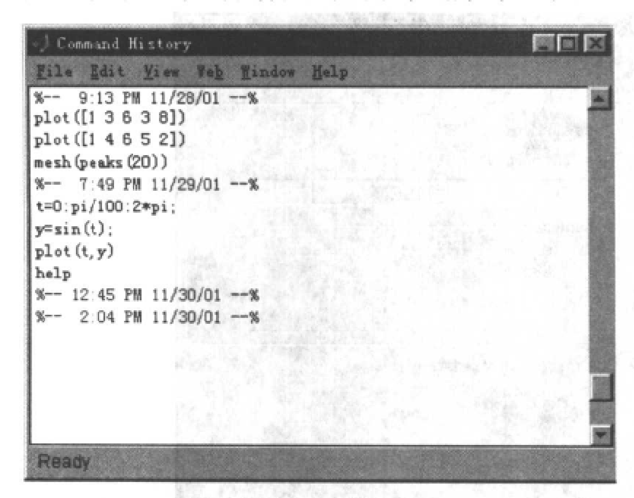


图 1-17　命令历史窗口

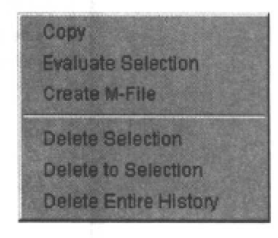


图 1-18　命令历史窗口弹出菜单

表 1-5 命令历史窗口弹出菜单的选项

菜单项	功能
Copy	将所选历史命令拷贝到剪贴板
Evaluate Selection	运行所选历史命令
Create M-File	打开编辑器，将所选历史命令复制到编辑器
Delete Selection	删除所选历史命令（不确认是否删除）
Delete to Selection	从头删除到所选历史命令（不确认是否删除）
Delete Entire History	清除全部历史命令（将弹出对话框确认是否清除）

1.3.8 当前路径窗口

1. 当前路径窗口及弹出菜单

MATLAB 6 的当前路径（Current Directory）窗口提供了当前路径下文件的操作（图 1-19），在该窗口的某一文件上单击鼠标右键会弹出菜单（图 1-20），菜单的选项见表 1-6。

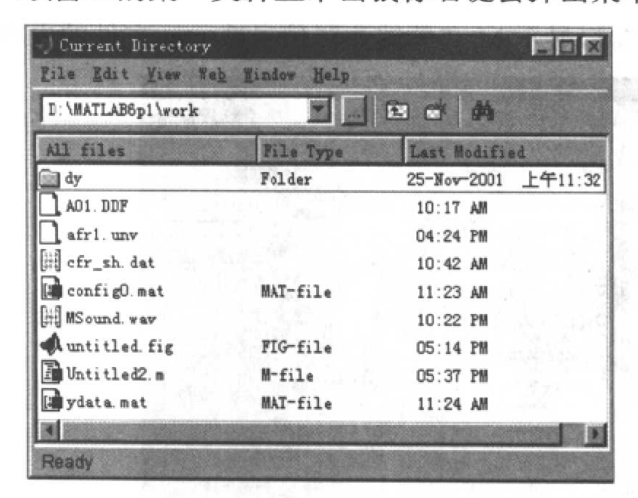


图 1-19 当前路径窗口

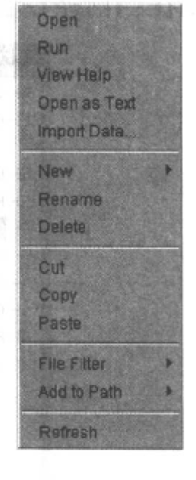


图 1-20 当前路径窗口弹出菜单

表 1-6 当前路径窗口弹出菜单的选项

菜单项	功能
Open	打开所选文件(M 文件在编辑器打开，mat 文件调入工作空间，fig 文件显示图形用户界面设计，其他数据文件用输入向导 Import Wizard 打开)
Run	执行 M 文件，如果所选文件不是 M 文件，该菜单不激活
View Help	在帮助浏览器中显示 M 文件头部的注释，如果所选文件不是 M 文件，该菜单不激活
Open as Text	在编辑器中编辑所选文件
Import Data…	导入数据（同主菜单）
New	新建（可以建立 M 文件、Simulink 模块或子目录）
Rename	文件更名
Delete	删除所选文件（将弹出对话框确认是否删除）

续表

Cut	剪切文件
Copy	复制文件
Paste	粘贴文件
File Filter	文件过滤器（缺省为显示全部文件和子目录）
Add to Path	添加到路径
Refresh	刷新显示

2. 当前路径窗口参数设置

当前路径窗口的参数设置如图 1-21 所示。

• History（历史）设置所保存的历史路径个数。Clear History（清除历史）按钮清除全部历史路径。

• Browse display options（路径显示选项）的 4 个选项分别为“Show file types”（显示文件类型）、“Show last modified date”（显示最后修改日期）、“Show M-file descriptions”（显示 M 文件注释行的第一行）、“Show M-file comments and MAT-file content”（如果勾选该项则在当前路径窗口中增加一个窗口显示 M 文件的注释信息或 mat 文件的内容）。

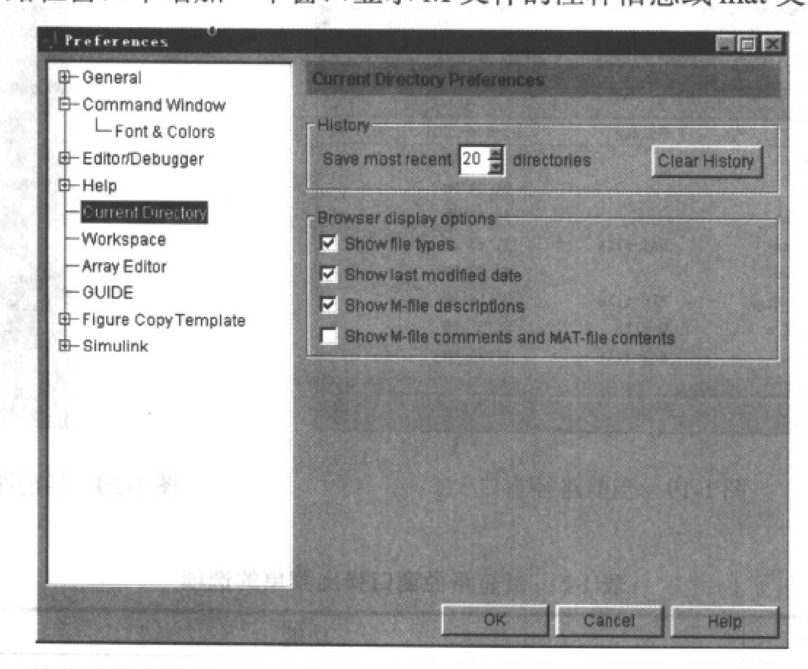


图 1-21　当前路径窗口的参数设置

1.3.9　MATLAB 的搜索路径

搜索路径是 MATLAB 定义的一系列文件路径的组合，缺省状态下包括当前路径和已安装的全部工具箱的路径，当程序或命令执行时，MATLAB 在搜索路径中查找命令或程序运行所需的 M 文件和各种数据文件。MATLAB 执行搜索时按照规定的顺序。以命令窗口为例，当输入一个字符串时，MATLAB 按下列顺序开始搜索：

(1) 在工作空间中搜索是否有以该字符串为变量名的变量，如果有，返回该变量的值；

(2) 搜索是否有同名的 MATLAB 内部函数，如果有，执行该内部函数；

(3) 在当前目录中查找与该字符串同名的 M 文件，如果有，执行该文件；

(4) 在搜索目录中查找与该字符串同名的 M 文件，如果有，执行该文件。

如果在搜索目录中存在多个同名函数，则只执行搜索路径中的第一个函数，其他函数不再执行。

搜索目录的设置通过选择主菜单的“Set Path...”菜单项，打开搜索路径设置窗口（图 1-22）。在右边的列表框中列出了全部的搜索路径，可以使用“Move to Top”、“Move Up”、“Move Down”、“Move to Bottom”等按钮调整搜索路径的顺序，Remove 按钮删除选中的路径。

使用“Add Folder...”按钮将某一目录加入搜索路径，如果选择“Add with Subfolders...”按钮，则将选中目录的子目录也包括在搜索路径中。

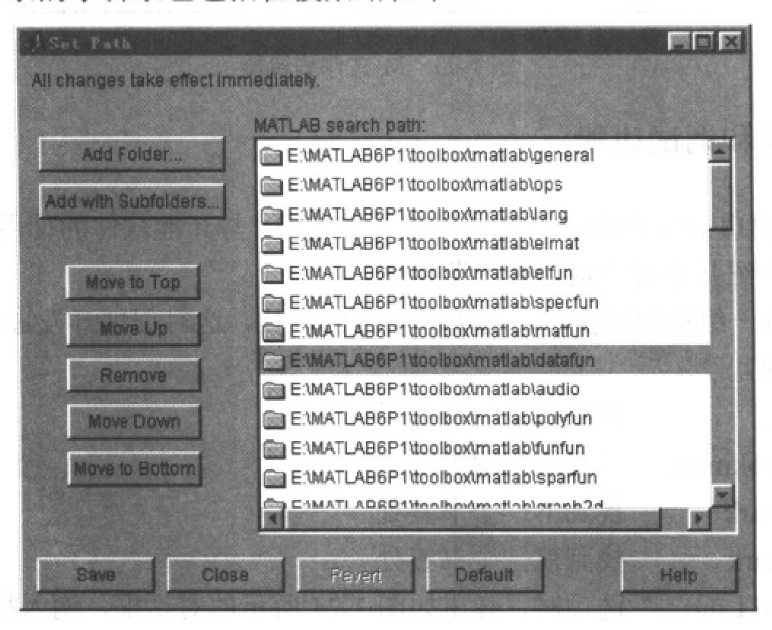


图 1-22 搜索路径设置窗口

1.4 M 文件编辑器

1.4.1 编辑器及界面

MATLAB 的命令文件和函数文件都是扩展名为“.m”的文件，通常称之为 M 文件。M 文件是纯文本的文件，可以用任何文本编辑器编辑，但 MATLAB 的开发环境中包括了专门的 M 文件编辑器，该编辑器不但提供了 M 文件的编辑功能，同时还与 MATLAB 的开发环境一起实现了 MATLAB 命令和函数文件的运行和调试。新建或打开一个 MATLAB 的命令或函数文件时，M 文件编辑器自动打开（图 1-23）。

M 文件编辑器的菜单栏和工具栏下面有 3 个区域，右侧的大块区域是程序窗口，用于编写程序；最左面的区域显示行号，每行都有数字，包括空行，行号是自动出现的，随着行数的增加而增加；在行号和程序窗口之间的区域上有一些小横线，这些横线只有在可执行行上才有，空行、注释行、函数定义行等前面都没有，在进行程序调试时，可以直接在这些横线

上点击鼠标以设置或去掉断点。

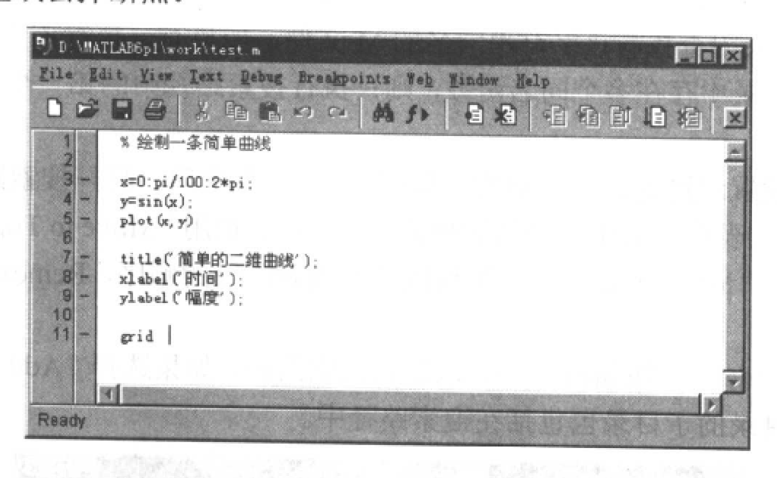


图 1-23　M 文件编辑器

1.4.2　编辑功能和调试功能

在编辑/调试窗口的菜单中有许多通用的菜单选项，与其他软件的使用方法相似，这里不逐一介绍，下面介绍几个有特定功能的菜单项。

Edit 菜单中有一组特别的操作，用于程序行的查找，这对程序开发过程中内容的查找和修改是非常有用的。

- Go to Line…　　　　找到指定行号处
- Set/Clear Bookmark　设置或清除书签
- Next Bookmark　　　找到下一个书签处
- Prev Bookmark　　　找到前一个书签处

当选定“Go to Line…”选项时，弹出如图 1-24 所示的对话框，输入行号并选择 OK 后，光标自动移到指定的行上，并将该行作为当前行。

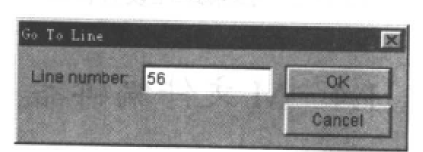


图 1-24　“Go to Line…”对话框

“Set/Clear Bookmark”选项用来设置或清除书签。首先将光标移到想要设置书签的行上，然后选择该项，在光标所在的程序行前面就会出现一个篮色的矩形标记，如图 1-25 所示，这就是 MATLAB 中 M 文件的书签标记。一个程序中可以设置多个书签标记，通过选择菜单中的“Next Bookmark”选项或“Prev Bookmark”选项，可以找到光标所在位置的后一个或前一个书签所在的行。如果要清除书签标记，只要将光标移到要清除书签的程序行上，再选择一次“Set/Clear Bookmark”选项，书签标记就消失了。

设置书签的目的是为了查找程序中某些内容所在的位置，书签只按程序内容标识，而不是按行号标识，如果在书签前面的位置添加或删除程序行时，书签标记会随着程序的内容而移动，这一点与通过行号去查找程序的方法是不同的。

MATLAB 的程序设计中规定，如果一程序行中的第一个符号是百分号“%”，则该行

为注释行，注释行只用来说明而不能执行。Text 菜单中的 Comment 选项的功能可以将选定的行设置为注释行，即在所有选定的行前面都加上一个“%”（无论所选定的行是可执行行、空行或注释行等），而 Uncomment 选项则在选定行中去掉一个百分号符号。

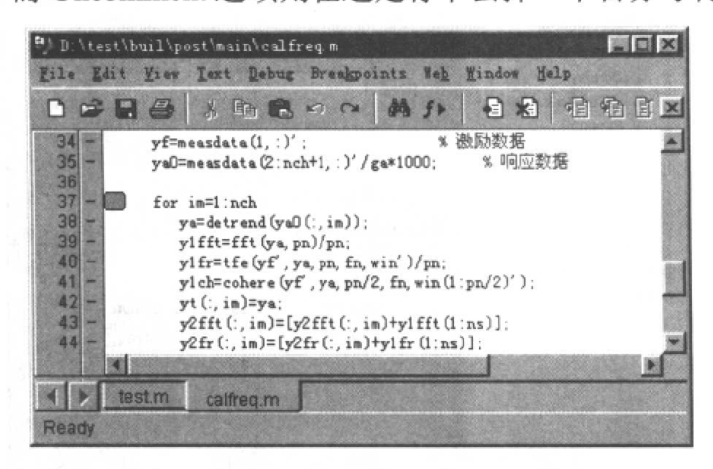


图 1-25　设置书签标记

Debug（调试）和 Breakpoints（断点）菜单是用来进行程序调试的，两者需要联合使用。

Debug 菜单中共有 6 个菜单项，在程序运行之前，仅有“Save and Run”项是激活的，只有当程序运行到第一个断点处并暂停时，其他菜单项方被激活，这些菜单项为：

- Step　单步运行，快捷键为[图标]。
- Step in　单步运行，遇到函数时进入函数，仍单步运行，快捷键为[图标]。
- Step out　如果是在函数中，跳出函数；如果不在函数中，直接运行到下一个断点处，快捷键为[图标]。
- Save and Run　存储文件并开始运行，如果文件是已经存储过的，该菜单项变为 Run，当程序暂停在断点处时，该菜单项变为 Continue，快捷键为[图标]。
- Go Until Cursor　直接运行到光标所在的位置。
- Exit Debug Mode　退出调试方式，快捷键为[图标]。

Breakpoints 菜单中共有 6 个菜单项，前两个用于在程序中设置和清除断点：

- Set/Clear Breakpoint　设置或清除断点，快捷键为[图标]。
- Clear All Breakpoint　清除所有断点，快捷键为[图标]。

后 4 个菜单项用于设定停止条件，用于临时停止 M 文件的执行，给用户一个检查局部变量的机会，相当于在文件编辑过程中在指定的行号前加入一个 keyboard 命令：

- Stop If Error　出现错误时停止程序运行，不包括 try...catch 语句中的错误。
- Stop If Warning　出现警告时停止程序运行。
- Stop If NaN or Inf　出现非数或无穷大时停止程序运行。
- Stop If All Error　与“Stop If Error”相同，但包括 try...catch 语句中的错误。

注：try...catch 语句的具体用法参见 8.3.4。

1.4.3　M 文件编辑器的参数设置

选择 File 菜单中的“Preferences...”选项打开参数设置窗口，选择树状列表中的

“Editor/Debugger”项（图 1-26），在右侧窗口中可以选择：

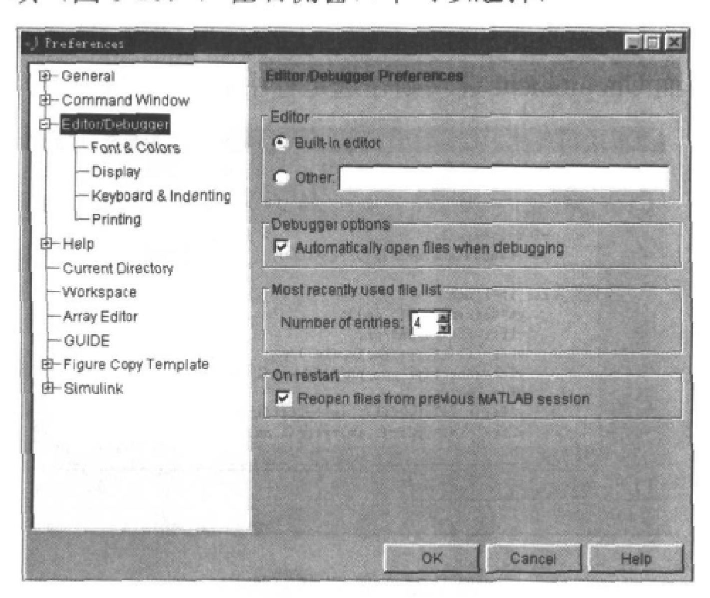


图 1-26　M 文件编辑器参数设置窗口

• Editor（文件编辑器）　选择“Built-in editor”（内部编辑器）或 Other（其他编辑器，在输入框中输入编辑器的路径和应用文件名）。

• Debugger options（调试器选项）　勾选（或不选）“Automatically open files when debugging”（在调试过程中自动打开遇到的文件）。

• Most recently used file list（近期使用过的文件列表）　在数字框中输入数字（不能超过 9）。

• On restart（重新启动）　勾选（或不选）“Reopen files from previous MATLAB session”（重新启动 MATLAB 系统时打开上次关闭系统时未关闭的文件）。

在树状结构中展开“Editor/Debugger”项，可以打开另外 4 个设置参数的窗口：

(1) Font & Colors（字体和颜色）（如图 1-27）

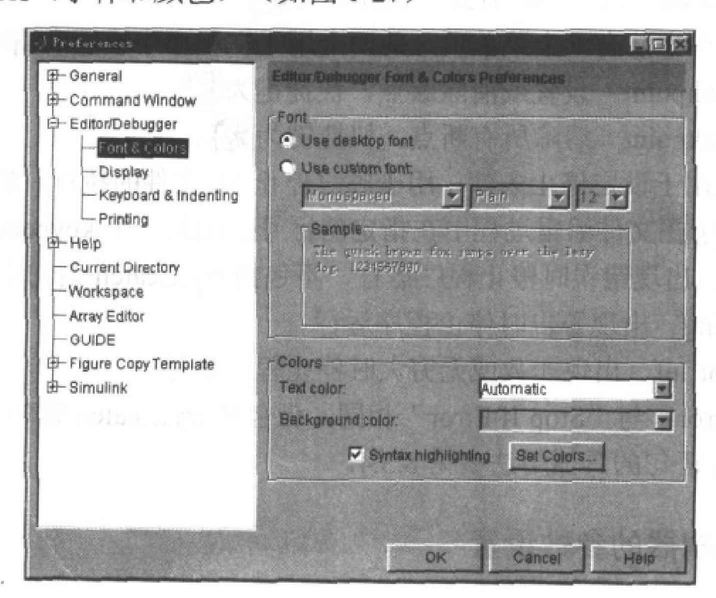


图 1-27　M 文件编辑器-字体和颜色设置窗口

• Font（字体） 选择“Use desktop font”（使用桌面字体）或“Use custom font”（使用用户字体）。当选择后者时，字体、字型、字号和示例框激活。

• Colors（颜色） “Text Color”（文本颜色），“Background Color”（背景颜色），选择下面的“Syntax highlighting”可以使语法语句带有不同的颜色，缺省状态下为：关键字（如 for…end、if…else…end、switch…case…end、function 等）为篮色，注释为绿色，字符串为棕色，不完整的字符串为淡紫色，系统命令为土黄色，错误为红色。按“Set Colors”按钮可以重新设置颜色。

(2) Display（显示）（如图 1-28）

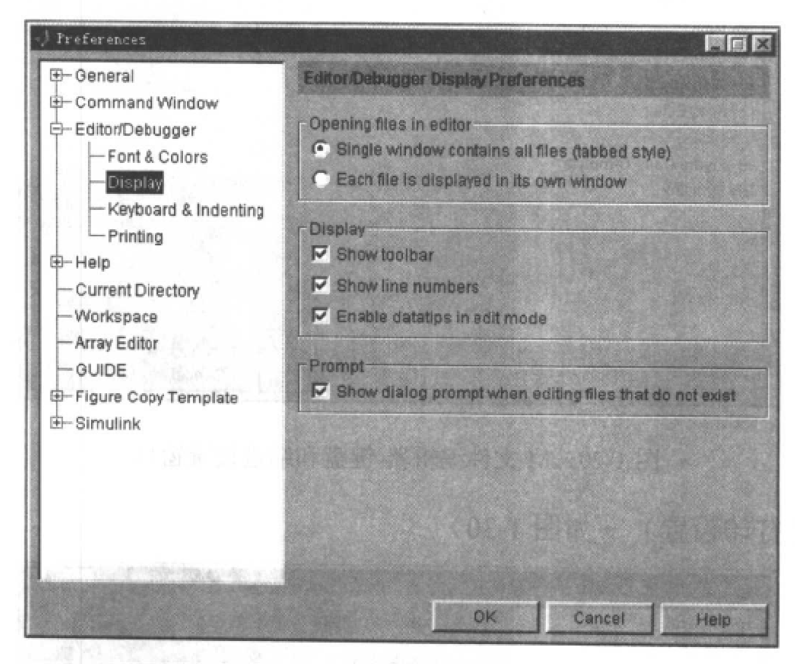


图 1-28 M 文件编辑器-显示设置窗口

• Opening files in editor（在编辑器中打开文件） 选择“Single window contains all files”（在一个窗口中显示所有的文件，文件名以页面标签形式显示在下方）或“Each file is displayed in its own window”（每个文件都在自己的窗口显示）。

• Display（显示） “Show toolbar”（显示工具栏）、“Show line numbers”（显示行号）和“Enable datatips in edit mode”（是否选择数据提示方式。如果选择数据提示方式，当执行完程序或是在调试过程中暂停时，将鼠标移至程序中的某一变量处时可以显示出该变量的内容）。

• Prompt（提示） “Show dialog prompt when editing files that do not exist”（当编辑的文件不存在时显示提示对话框）。

(3) Keyboard & Indenting（键盘和缩进）（如图 1-29）

• Key bindings（键盘的定义习惯） Windows（Windows 定义习惯），Emacs（Emacs 定义习惯）。

• M-file indenting for Enter key（M 文件的缩进设置） “No indent”（无缩进）、“Block indent”（以块的形式缩进）或“Smart indent”（智能缩进）。

• Intents（缩进设置） 在“Indent size”（缩进尺寸）输入框中输入数字，选择“Emacs

style Tab key smart indenting”（Emacs 风格时 Tab 键智能缩进）。

• Tabs（Tab 键设置）　在“Tab size”（Tab 尺寸）输入框中输入数字，选择“Tab key inserts spaces”（在输入 Tab 键时是插入一个 Tab 还是插入等量的空格）。

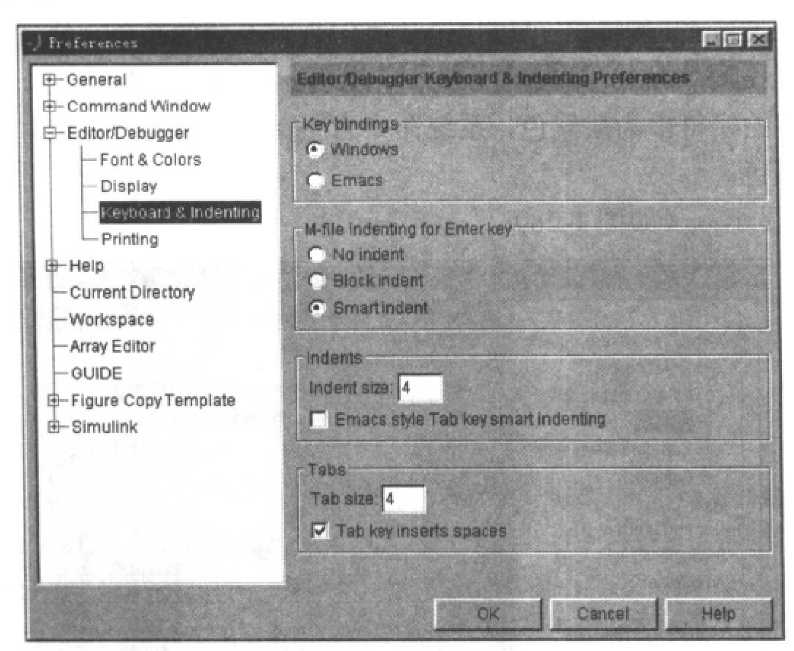


图 1-29　M 文件编辑器-键盘和缩进设置窗口

(4) Printing（打印设置）（如图 1-30）

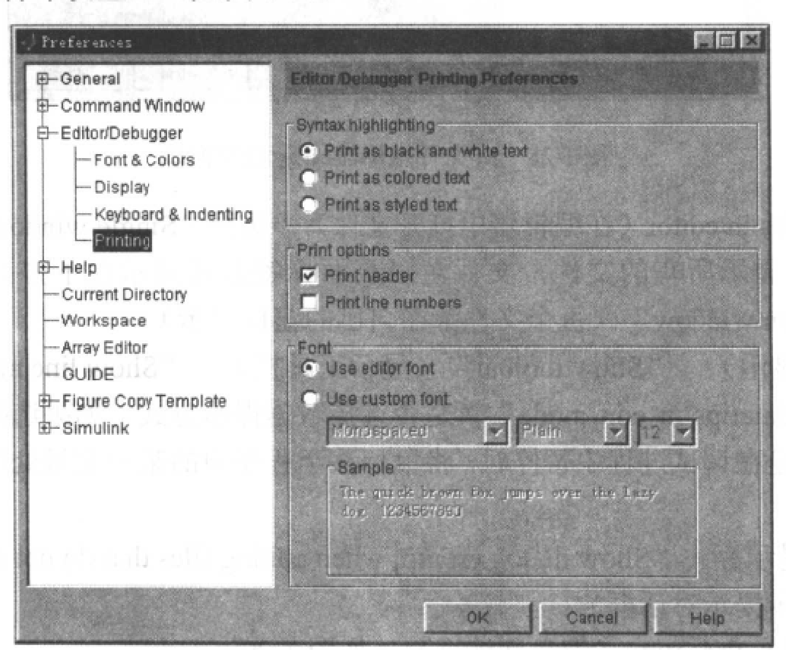


图 1-30　M 文件编辑器-打印参数设置窗口

• Syntax highlighting（打印带色彩的语句时的选择）　“Print as black and white text”（按黑白文本方式打印），“Print as colored text”（按彩色文本方式打印），“Print as styled text”（按设置的风格打印文本）。

• Print options（打印选项）　“Print header”（是否打印文件头，如路径、文件名等）、

"Print line numbers"（是否打印行号）。

• Font（字体选择） "Use editor font"（使用编辑器的字体），"Use custom font"（使用用户定义的字体）。当选择后者时，下面的字体、字型和字号的选择框激活。

1.5 在线帮助

MATLAB 提供了功能强大的在线帮助系统，用户可以随时通过在线帮助系统获得帮助信息，MATLAB 中可以使用多种方法获得帮助信息。

1.5.1 帮助浏览器

帮助浏览器是 MATLAB 6 新增加的功能。当选择了 View 菜单中的 Help 选项、在启动平台的树状列表中选择了 Help 项或在 Help 菜单中选择 MATLAB Help 菜单项时，MATLAB 打开一个独立的交互式帮助浏览器，如图 1-31 所示。

帮助浏览器的左侧有 4 个重叠的页面：第 1 页 Contents（目录）是一个树状列表，按内容列出了所有的帮助主题，点击相应的内容就会在右侧的超文本浏览器中显示出详细的帮助内容；第 2 页 Index（索引）按字母顺序列出帮助条目，同时指出每一条目内容所在的位置（如属于基本平台中的某一部分或属于某一工具箱等），先按字母顺序查找到需要的条目，点击该条目就会在右侧的超文本浏览器中显示帮助内容；第 3 页 Search（搜索）允许用户搜索指定内容，在 Search for 栏中输入要搜索的关键词后按 Go 按钮，下面窗口显示出搜索到的条目，再点击其中某一具体条目，就可以在右侧的超文本浏览器中显示出详细的帮助内容；第 4 页 favorites（常用的内容）中存储了用户常用的内容（以便于快速查找），存储第 4 页的方法：在超文本浏览器的上方显示当前帮助内容的主题，按"Add to favorites"按钮，将该主题添加到第 4 页中，也可以在左边窗口中该主题上按鼠标右键，弹出一个标有"Add to favorites"字样的菜单，点击该菜单就将该主题添加到第 4 页中。

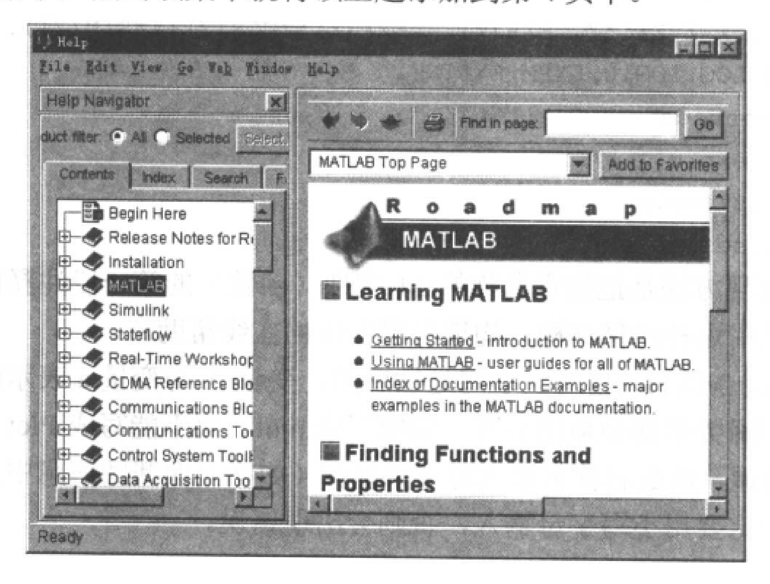


图 1-31 帮助浏览器

1.5.2 help 命令

在命令窗口输入 help 命令，是 MATLAB 在线帮助的传统形式，也是一种方便而快捷的方法，help 命令的用法为：

- help　　　　　列出主要的帮助主题。
- help 帮助主题　列出指定主题下的函数。
- help 函数名　　对所选函数的功能、调用格式及相关函数给出说明。

【例 1-3】 显示在线帮助总览。

在 MATLAB 的命令窗口，输入 help，系统将显示：

```
>> help
HELP topics:
    matlab\general   - General purpose commands.
    matlab\ops       - Operators and special characters.
    matlab\lang      - Programming language constructs.
    matlab\elmat     - Elementary matrices and matrix manipulation.
    matlab\elfun     - Elementary math functions.
    matlab\specfun   - Specialized math functions.
    ...
    MATLAB\work      - (No table of contents file)
    toolbox\local    - Preferences.
    For more help on directory/topic,type "help topic".
```

【例 1-4】 显示指数函数命令 exp 的详细信息。

在 MATLAB 的命令窗口，输入 help exp，系统将显示：

```
>> help exp
EXP Exponential.
    EXP(X) is the exponential of the elements of X,e to the X.
    For complex Z=X+i*Y, EXP(Z) = EXP(X)*(COS(Y)+i*SIN(Y)).
    See also LOG,LOG10,EXPM,EXPINT.
Overloaded methods
    help sym/exp.m
    help fints/exp.m
    help demtseries/exp.m
```

help 命令的实现方法是把指定名称的 M 文件（函数）的第一段注释内容显示出来。因此，用户也可以采用这种注释结构，构造自己文件的在线帮助。

需要注意的是，MATLAB 对大小写是敏感的，变量 A 与变量 a 表示的是两个不同的变量，所有的命令和函数名都必须用小写（如应写成 plot，而不能写成 Plot 或 PLOT）。在帮助信息中，为了强调，函数名称都用大写字母显示（如在上面帮助信息中显示的 EXP），但在实际运行时，一定要用小写字母输入，否则会出现错误。

1.5.3 lookfor 命令

当希望查找具有某种功能的函数但不知道该函数准确名称时，help 就无能为力了。为此，MATLAB 设计了另外一个命令 lookfor，它可以根据用户提供的完整或不完整的关键词，搜索与该关键词有关的函数或命令。

【例 1-5】 查找有关积分的函数，使用 integral 作为关键词。

在 MATLAB 的命令窗口，输入 lookfor integral，系统将显示全部与积分有关的函数：

```
>> lookfor integral
ELLIPKE      Complete elliptic integral.
EXPINT       Exponential integral function.
DBLQUAD      Numerically evaluate double integral.
QUAD         Numerically evaluate integral,adaptive Simpson quadrature.
...
```

1.5.4 模糊查询

MATLAB 6 以后的版本提供了另一种方便的查询方法，即模糊查询方法。使用模糊查询方法，用户可以只输入命令的前几个字母，然后按 Tab 键，系统就会列出全部以这几个字母开始的函数。如键入 plot 后按 Tab 键，则列出所有以 plot 开头的函数：

```
plot      plotbr     plotes    plotmap      plotperf    plottr
plot3     plotchar   plotfa    plotmatrix   plotpv      plotv
plot3m    ploteach   plotfis   plotmf       plotscale   plotvec
...
```

这样用户就知道某一函数的确切名称，然后再通过其他方式查询其详细的功能和用法。

1.5.5 在线帮助的参数设置

在线帮助的参数设置如图 1-32 所示。

• Documentation location（帮助文件位置） 有两个选项，分别为“Local or network directory”（本地或网络目录）和“CD-ROM drive”（光驱）。如果安装了帮助文件，选前者，否则选后者。

• Product filter（产品过滤器） 通过“Select products...”按钮选择在帮助浏览器中列出的帮助主题。要注意的是，在帮助浏览器的左上方有同样功能的“Select...”按钮和两个单选框，过滤器只有在选择 Selected 单选框时才起作用。

• General（常规） 勾选“Keep contents tree synchronized with displayd document”项控制帮助浏览器的同步特性。

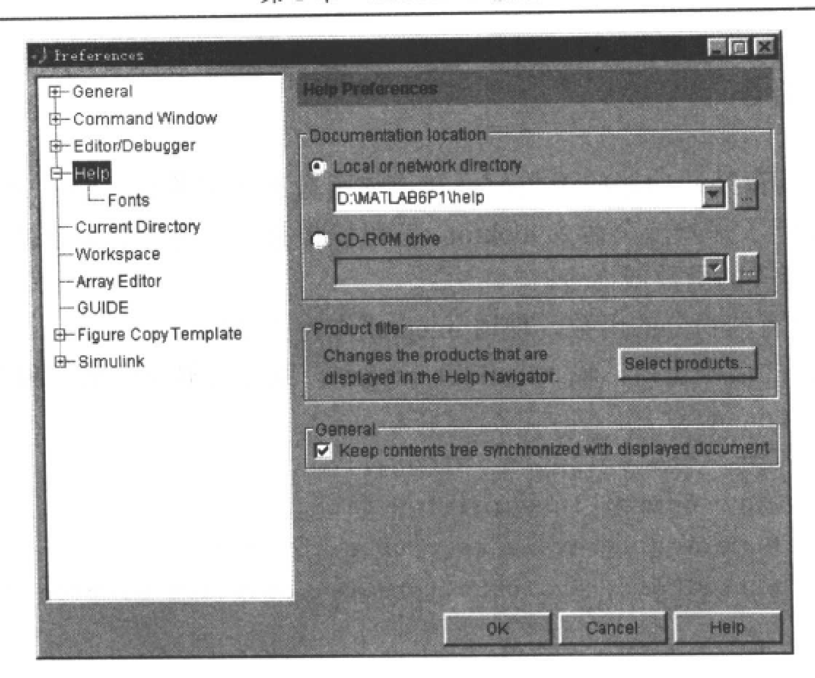


图 1-32　在线帮助设置窗口

在树状结构中展开 Help 项，可以设置 Font（字体）（图 1-33）。上部设置帮助浏览器的字体，缺省状态为使用操作桌面的字体，也可以单独设置字体；下部设置超文本浏览器的字体。

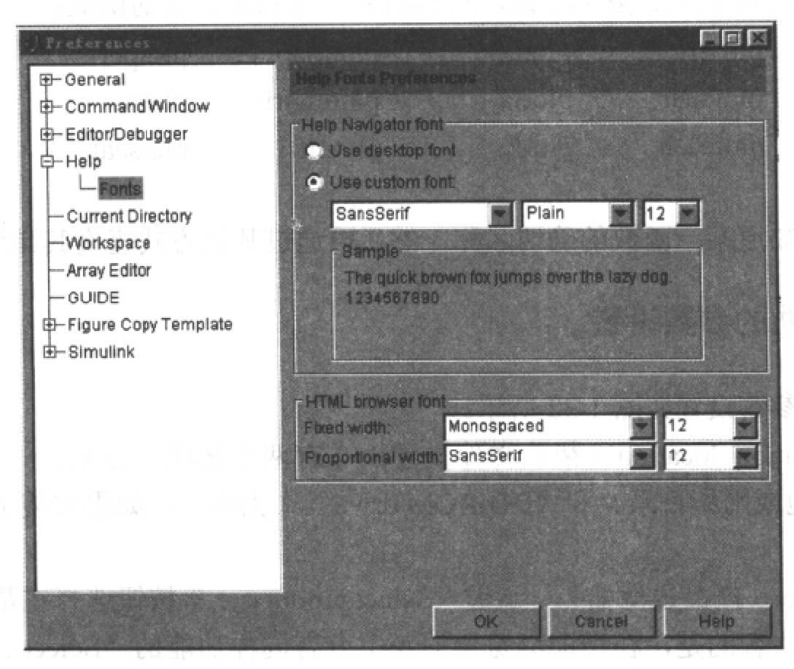


图 1-33　在线帮助 字体和颜色设置窗口

习　题　1

1.1　与其他计算机语言相比较，MATLAB 语言突出的特点是什么？

1.2　MATLAB 系统由哪些部分组成？

1.3 安装 MATLAB 时，在选择组件窗口中哪些部分必须勾选，没有勾选的部分以后如何补装？

1.4 MATLAB 操作桌面有几个窗口？如何使某个窗口脱离桌面成为独立窗口？又如何将脱离出去的窗口重新集成到桌面上？

1.5 如何启动 M 文件编辑/调试器？

1.6 存储在工作空间中的数组能编辑吗？如何操作？

1.7 命令历史窗口除了可以观察已经输入的命令外，还有什么用途？

1.8 如何设置当前目录和搜索路径，在当前目录上的文件和在搜索路径上的文件有什么区别？

1.9 MATLAB 中有几种获得帮助的途径？

第 2 章　矩阵运算基础

MATLAB 作为优秀的数学计算和分析软件，以矩阵运算为代表的基本运算功能一直是 MATLAB 引以为自豪的核心和基础。在 MATLAB 中，标量可以看成是 1×1 的矩阵，*n* 维矢量可以看成 *n*×1 的矩阵，多项式可以由它的系数矩阵完全确定。在 MATLAB 中，矩阵和数组的输入形式和书写方法是相同的，不需要加以说明，都是一些数的集合，如 *a*=[1 2;3 4]，*b*=[1 3 5;2 4 6]等，其区别仅仅在于进行运算时：数组的运算是数组中对应元素的运算，而矩阵运算则应符合矩阵运算的规则。本章将详细介绍 MATLAB 的基本运算功能，包括矩阵和数组的创建及其运算方法。

2.1　矩阵的创建

在 MATLAB 中创建矩阵应遵循以下原则：

- 矩阵的元素必须在方括号“[ ]”中；
- 矩阵的同行元素之间用空格或逗号“，”分隔；
- 矩阵的行与行之间用分号“；”或回车符分隔；
- 矩阵的尺寸不必预先定义；
- 矩阵元素可以是数值、变量、表达式或函数。

在 MATLAB 中，矩阵的创建有 4 种方法，分别介绍如下。

2.1.1　命令窗口直接输入

在命令窗口直接输入矩阵是最方便简洁的矩阵创建方法，只要遵循矩阵创建的原则直接输入矩阵元素，如果不希望显示结果，在命令行的最后加分号“；”（该原则适用于 MATLAB 的所有操作，包括程序设计）。

【例 2-1】　在命令窗口创建简单的数值矩阵。

在命令窗口输入：

```
>> x=[1 2 3;4 5 6;7 8 9]
x =
     1     2     3
     4     5     6
     7     8     9
```

【例 2-2】　在命令窗口创建带运算表达式的矩阵，不显示结果。

在命令窗口输入：

```
>> y=[sin(pi/3),cos(pi/6);log(20),exp(2)];
```

系统创建了一个 2×2 的矩阵，但没有在命令窗口中显示结果，用户可以通过工作空间

窗口查看。

2.1.2 通过 M 文件创建矩阵

当矩阵的尺寸较大时，直接在命令窗口输入矩阵元素就显得笨拙，容易出现错误且不便修改。为了解决此问题，可以先将矩阵按创建原则写入一个 M 文件中，在 MATLAB 的命令窗口或程序中直接执行该 M 文件，即将矩阵调入工作空间。

2.1.3 通过函数创建矩阵

对于一些特殊的矩阵，可以利用 MATLAB 的内部函数或用户自定义函数创建矩阵。

【例 2-3】 创建 0 到 2π 间的正弦函数矩阵。

在命令窗口输入：

```
>> x=0:pi/4:2*pi;
```

表示创建了 0 到 2π 间间隔为$\pi/4$ 的自变量，然后输入：

```
>> y=sin(x)
```

得到 0 到 2π 间间隔为$\pi/4$ 的正弦函数值。

```
y =
  Columns 1 through 6
        0      0.7071     1.0000     0.7071     0.0000     -0.7071
  Columns 7 through 9
       -1.0000 -0.7071    -0.0000
```

2.1.4 通过数据文件创建矩阵

在很多应用问题中，可能会通过许多不同的方式取得数据，例如，从某一计算程序算得的结果或从某一实验中测得的数据等。这些数据一般可以以 ASCII 码的格式提供，使用 MATLAB 可以方便地处理这些数据。

事实上，MATLAB 可以处理的数据格式很多，除常用的文本文件、MATLAB 的.mat 数据文件、Excel 的数据表.xls 文件外，MATLAB 还可以处理大多数图像文件和声音文件。在工作空间中，这些文件都是作为矩阵存储的。

MATLAB 既可以在命令窗口或通过编制程序调入各种文件，同时又提供了功能强大的数据导入向导（Import Wizard）调入各种数据。使用数据导入向导功能和编程调入数据的方法将在 8.2.3 和 8.2.4 中介绍。

2.2 矩阵和数组的算术运算

2.2.1 矩阵和数组的加减运算

矩阵和数组的加减运算没有区别，其运算法则与普通的加减运算相同，但要注意相加减的两个矩阵必须有相同的阶数，除非其中的一个为标量（即常数）。

【例 2-4】 两个矩阵分别为$\begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\\7 & 8 & 9\end{bmatrix}$和$\begin{bmatrix}1 & 1 & 1\\2 & 2 & 2\\3 & 3 & 3\end{bmatrix}$，求两者相加的和。

在命令窗口输入，并计算：

```
>> a=[1 2 3;4 5 6;7 8 9];
>> b=[1 1 1;2 2 2;3 3 3];
>> c=a+b
```

结果为：

```
c =
    2     3     4
    6     7     8
    10    11    12
```

【例 2-5】 两个矩阵分别为$\begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\\7 & 8 & 9\end{bmatrix}$和$\begin{bmatrix}1 & 1 & 1\end{bmatrix}$，阶数不同，求两者相减的差。

在命令窗口中输入，并计算：

```
>> a=[1 2 3;4 5 6;7 8 9];
>> b=[1 1 1];
>> c=a - b
```

则结果显示：

```
??? Error using ==> -
Matrix dimensions must agree.
```

2.2.2 矩阵的乘法

矩阵的乘法使用“*”运算符，要求相乘的矩阵有相邻的公共阶，即当矩阵 a 为 $i\times j$ 阶，矩阵 b 为 $j\times k$ 阶时，矩阵 a 和 b 才能相乘。

【例 2-6】 两个矩阵相乘，矩阵 a 为$\begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\\7 & 8 & 9\end{bmatrix}$，矩阵 b 为$\begin{bmatrix}1 & 2 & 3\end{bmatrix}$。

在命令窗口输入两矩阵，并分别计算 $c=a*b$ 和 $d=b*a$：

```
>> a=[1 2 3;4 5 6;7 8 9];
>> b=[1 2 3];
>> c=a*b
??? Error using ==> *
Inner matrix dimensions must agree.
>> d=b*a
d =
    30    36    42
```

2.2.3 数组的乘法

数组的乘法用符号“.*”表示，a、b 两数组必须具有相同的阶数，则 $a.*b$ 表示 a 和 b

中对应元素之间相乘。

【例 2-7】 两个数组相乘，数组 a 为 $[1\ \ 2\ \ 3]$，数组 b 为 $[4\ \ 5\ \ 6]$，求两数组的乘法。

在命令窗口输入两数组，计算 $c = a.*b$：

```
>> a=[1 2 3];
>> b=[4 5 6];
>> c=a.*b
c =
     4    10    18
```

即 $c(i,j) = a(i,j)*b(i,j)$。

2.2.4 矩阵的除法

MATLAB 中的矩阵除法有两种，分别为左除“\”和右除“/”，如果 a 是非奇异方阵，则 $a\backslash b$ 和 b/a 的运算都可以实现。$a\backslash b$ 等效于矩阵 a 的逆左乘矩阵 b，即 inv(a)*b，而 b/a 等效于矩阵 a 的逆右乘矩阵 b。

在通常情况下，$x = a\backslash b$ 是 $a*x = b$ 的解，$x = b/a$ 是 $x*a = b$ 的解。一般情况下，$a\backslash b \neq b/a$。

左除与右除的关系为：$b/a = (a'\backslash b')'$。

【例 2-8】 两个矩阵相除，矩阵 a 和 b 均为 3×3 阶矩阵。

在命令窗口利用 rand 函数创建两矩阵，并分别计算 $c = a/b$ 和 $d = b\backslash a$：

```
>> a=rand(3)
a =
    0.9501    0.4860    0.4565
    0.2311    0.8913    0.0185
    0.6068    0.7621    0.8214
>> b=rand(3)
b =
    0.4447    0.9218    0.4057
    0.6154    0.7382    0.9355
    0.7919    0.1763    0.9169
>> c=a/b
c =
    1.7993   -2.0059    1.7482
    1.3530   -0.4548   -0.1144
    0.2744    0.6673    0.0936
>> d=b\a
d =
    3.2912   -0.0601    3.4561
    0.4400    0.1830   -0.2760
   -2.2653    0.8479   -2.0362
```

2.2.5 数组的除法

数组除法用符号“.\”或“./”表示（两者的结果相同），a 与 b 必须具有相同的阶数。

$a.\backslash b$ 表示 b 中的元素分别除以 a 中的对应元素。

【例 2-9】 数组 a 为$[1\ \ 2\ \ 3]$，数组 b 为$[4\ \ 5\ \ 6]$，求两数组的除法。

在命令窗口输入两数组，计算 $c=a.\backslash b$：

```
>> a=[1 2 3];
>> b=[4 5 6];
>> c=a.\b
c =
    4.0000    2.5000    2.0000
>> c=b./a
c =
    4.0000    2.5000    2.0000
```

即 $c(i,j)=a(i,j)\backslash b(i,j)=b(i,j)/a(i,j)$。

注意，这里 $a.\backslash b$ 和 $b./a$ 一样，得到相同的结果（b 除以 a）。与矩阵的左除、右除是不一样的，它只是两数组对应元素间的运算。

2.2.6 矩阵的幂运算

矩阵幂运算用“^”符号来表示，a^p 的意思是 a 的 p 次方。

如果 a 是一个方阵，p 是一个标量，且 p 是大于 1 的整数，则 a 的 p 次幂即为 a 自乘 p 次。

如果 p 是不为整数的标量时，则 $a\text{^}p=V*\begin{bmatrix}\lambda_1^P & & \\ & \ddots & \\ & & \lambda_n^P\end{bmatrix}/V$，其中$\lambda_1$，…，$\lambda_n$为矩阵 a 的特征值，V 为对应的特征矢量阵，也可写成：$a\text{^}p=V*D.\text{^}p/V$，$[V,D]=\mathrm{eig}(a)$，eig 是计算特征值和特征矢量的函数，其结果可能是复数。

当 p 是方阵而 a 是标量时，计算也涉及到特征值和特征矢量的问题，$a\text{^}p=V*\begin{bmatrix}a^{\lambda_1} & & \\ & \ddots & \\ & & a^{\lambda_n}\end{bmatrix}/V$，其中$\lambda_1$，…，$\lambda_n$为矩阵 p 的特征值，V 为对应的特征矢量阵，也可写成：$a\text{^}p=V*a.\text{^}D/V$，$[V,\ D]=\mathrm{eig}(p)$。

如果 a、p 都是矩阵，则 a^p 是错误的，也就是说 a 和 p 中必须一个是标量，一个是方阵。

【例 2-10】 矩阵 a 为$\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}$，求它的 1.5 次幂。

在命令窗口输入矩阵，计算 $c=a\text{^}1.5$：

```
>> a=[1 2;3 4];
>> c=a^1.5
c =
    2.9746 - 0.1729i   4.3352 + 0.0791i
    6.5028 + 0.1186i   9.4774 - 0.0543i
```

2.2.7 数组的幂运算

数组的幂运算用符号“.^”表示，与数组的其他运算一样，数组的幂运算是对其元素逐一进行幂运算。

【例 2-11】 数组 a 为 $[1\quad 2\quad 3]$，数组 b 为 $[4\quad 5\quad 6]$，求数组的幂 $c=a.\hat{}b$。

在命令窗口输入两数组，计算 $c=a.\hat{}b$：

```
>> a=[1 2 3];
>> b=[4 5 6];
>> c=a.^b
c =
    1    32   729
```

本例执行的运算为：

$c=a.\hat{}b=[1\ 2\ 3].\hat{}[4\ 5\ 6]=[1^4\ 2^5\ 3^6]=[1\ 32\ 729]$

即 $c(i,j)=a(i,j)\hat{}b(i,j)$，为对应元素的幂运算。在这种底与指数均为数组的情况下，要求它们的阶数必须相同。

【例 2-12】 数组 a 为 $[1\quad 2\quad 3]$，求数组的幂 $c=a.\hat{}2$。

在命令窗口输入数组，计算 $c=a.\hat{}2$：

```
>> a=[1 2 3];
>> c=a.^2
c =
    1     4     9
```

本例执行的运算为：

$c=[1\ 2\ 3].\hat{}2=[1^2\ 2^2\ 3^2]=[1\ 4\ 9]$

即 $c(i,j)=a(i,j)\hat{}2$。

【例 2-13】 数组 a 为 $[1\quad 2\quad 3]$，求数组的幂运算 $c=2.\hat{}a$。

在命令窗口输入数组，计算 $c=2.\hat{}a$：

```
>> a=[1 2 3];
>> c=2.^a
c =
    2     4     8
```

本例执行的运算为：

$c=2.\hat{}a=2.\hat{}[1\ 2\ 3]=[2^1\ 2^2\ 2^3]=[2\ 4\ 8]$

即 $c(i,j)=2\hat{}a(i,j)$。

2.2.8 矩阵的转置

矩阵的转置用符号“'”来表示和实现。

【例 2-14】 矩阵 a 为 $\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{bmatrix}$，计算 a 的转置。

在命令窗口输入矩阵，计算 $c = a'$：

```
>> a=[1 2 3;4 5 6;7 8 9];
>> c=a'
c =
    1     4     7
    2     5     8
    3     6     9
```

如果输入矩阵 a 是复数矩阵，则 a'为它们的复数共轭转置，若要进行非共轭转置运算，使用 $a.'$或 conj(a')。

【例 2-15】 矩阵 a 为 $[1+2i \quad 3+4i]$，计算 a 的转置。

在命令窗口输入矩阵，计算 $c = a'$：

```
>> a=[1+2i 3+4i];
>> c=a'
c =
   1.0000 - 2.0000i
   3.0000 - 4.0000i
>> c=a.'
c =
   1.0000 + 2.0000i
   3.0000 + 4.0000i
>> c=conj(a')
c =
   1.0000 + 2.0000i
   3.0000 + 4.0000i
```

2.3 矩阵和数组的关系运算和逻辑运算

2.3.1 关系运算

MATLAB 提供了 6 种关系运算，其结果返回"1"或"0"，表示运算关系是否成立。关系运算符见表 2-1。

表 2-1 关系运算符

运算符	功能	运算符	功能
<	小于	>=	大于等于
<=	小于等于	==	等于
>	大于	~=	不等于

【例 2-16】 矩阵 a 和 b 均为 1×3 阶矩阵，使用关系运算符对对应元素进行比较。

在命令窗口输入两矩阵，然后比较两矩阵间对应元素的关系：

```
>> a=[0 -1 2];
```

```
>> b=[-3 1 2];
>> a<b
ans =
     0     1     0
>> a<=b
ans =
     0     1     1
>> a>b
ans =
     1     0     0
>> a>=b
ans =
     1     0     1
>> a==b
ans =
     0     0     1
>> a~=b
ans =
     1     1     0
```

关系运算符通常用于程序的流程控制中，常与 if、while、for、switch 等控制命令联合使用。

2.3.2 逻辑运算

在 MATLAB 中，有 3 种逻辑运算符用于逻辑运算，它们是"与"运算符"&"（或 AND）、"或"运算符"|"（或 OR）、"非"运算符"~"（或 NOT）。其中"&"和"|"是对同阶矩阵中的对应元素进行逻辑运算，如果其中一个是标量，则标量逐个与矩阵中的每一个元素进行逻辑运算。"~"用于对单个矩阵或标量进行取反运算。

- &（与）运算　当运算双方对应元素的值均为非 0 时，结果为 1，否则为 0;
- |（或）运算　当运算双方对应元素的值有一个为非 0 时，结果为 1，否则为 0;
- ~（非）运算　当元素的值为 0 时，结果为 1，否则为 0。

【例 2-17】　矩阵 a 和 b 均为 2×3 阶矩阵，使用逻辑运算符计算对应元素。

在命令窗口输入两矩阵，然后计算两矩阵对应元素的逻辑关系：

```
>> a=[1 0 3;0 -1 6];
>> b=[-1 0 0;0 5 0.3];
>> a&b
ans =
     1     0     0
     0     1     1
>> a|b
ans =
     1     0     1
     0     1     1
```

```
>> ~a
ans =
     0     1     0
     1     0     0
>> ~b
ans =
     0     1     1
     1     0     0
```

2.3.3　逻辑函数和关系函数

除了关系运算符和逻辑运算符外，MATLAB 还提供了更为方便的逻辑函数和关系函数，见表 2-2。

表 2-2　关系函数和逻辑函数

函数名称	功能	函数名称	功能
all(*x*)	检查 *x* 是否全为 1（TRUE）	isglobal(*x*)	检查 *x* 是否为全局变量
any(*x*)	检查 *x* 是否有不为 0 的元素	isinf(*x*)	检查 *x* 是否为无穷大
exist(*x*)	检查变量、函数或文件的存在性和类别	isnan(*x*)	检查 *x* 是否为 NaN
find(*x*)	找出非零元素的位置标识	issparse(*x*)	检查 *x* 是否为稀疏矩阵
isempty(*x*)	检查 *x* 是否为空阵	isstr(*x*)	检查 *x* 是否为字符串
isfinite(*x*)	检查 *x* 是否为有限值	xor(*x*, *y*)	执行异或运算（$x \otimes y$）

2.4　矩阵函数和数组函数

2.4.1　矩阵函数

MATLAB 提供了大量的矩阵函数，包括特征值的计算、奇异值的计算、条件数、范数、矩阵的秩和矩阵的空间运算等。表 2-3 中列出部分常用的矩阵函数，其他函数请查阅附录 A。

表 2-3　MATLAB 的矩阵函数

函数	功能	函数	功能
cond	矩阵的条件数	rank	矩阵的秩
condest	1-范数条件数	svd	矩阵的奇异值分解
rcond	矩阵 A 的倒条件数	trace	矩阵的迹
det	方阵的行列式	expm	矩阵指数
inv	矩阵的逆	logm	矩阵对数
pinv	矩阵的伪逆	sqrtm	矩阵平方根
norm	矩阵或矢量的范数	funm	矩阵的一般矩阵函数
normest	矩阵的 2-范数	eig	矩阵的特征值和特征矢量

【例 2-18】 矩阵 a 为 $\begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\\7 & 8 & 9\end{bmatrix}$，计算 a 的特征值和特征矢量。

在命令窗口输入矩阵，计算$[c, d] = \mathrm{eig}(a)$：

```
>> a=[1 2 3;4 5 6;7 8 9];
>> [c,d]=eig(a)
c =
   -0.2320   -0.7858    0.4082
   -0.5253   -0.0868   -0.8165
   -0.8187    0.6123    0.4082
d =
   16.1168         0         0
         0   -1.1168         0
         0         0   -0.0000
```

对于严重病态的矩阵，其逆函数可能是伪解，此时应使用伪逆函数。

【例 2-19】 矩阵 a 为 $\begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\\7 & 8 & 9\end{bmatrix}$，计算 a 的逆矩阵和伪逆矩阵。

在命令窗口输入矩阵，计算 $c = \mathrm{inv}(a)$ 和 $c = \mathrm{pinv}(a)$：

```
>> a=[1 2 3;4 5 6;7 8 9];
>> c=inv(a)
Warning: Matrix is close to singular or badly scaled.
         Results may be inaccurate. RCOND = 1.541976e-018.
c =
    1.0e+016 *
    -0.4504    0.9007   -0.4504
     0.9007   -1.8014    0.9007
    -0.4504    0.9007   -0.4504
>> c=pinv(a)
c =
   -0.6389   -0.1667    0.3056
   -0.0556    0.0000    0.0556
    0.5278    0.1667   -0.1944
```

2.4.2 通用函数

MATLAB 的通用函数包括数学函数和特殊函数两部分。

1. 数学函数

MATLAB 的基本数学函数分为 4 部分：三角函数、指数函数、复数函数、取整和求余函数。表 2-4 中只列出一些常用的函数，详细信息请查阅附录 A。

表 2-4　MATLAB 的基本数学函数

	函数名称	功能	函数名称	功能
三角函数	sin	正弦	sec	正割
	asin	反正弦	asec	反正割
	cos	余弦	csc	余割
	acos	反余弦	acsc	反余割
	sinh	双曲正弦	cot	余切
	cosh	双曲余弦	acot	反余切
	tan	正切	sech	双曲正割
	atan	反正切	csch	双曲余割
	atan2	第 4 象限的反正切		
指数函数	exp	以 e 为底的指数	pow2	2 的幂次
	log	自然对数	sqrt	开平方
	log10	以 10 为底的对数	nextpow2	2 的下一个幂次
	log2	以 2 为底的对数		
复数函数	abs	绝对值或复数的模	real	复数的实部
	angle	相位角	unwrap	相位展开
	complex	由实部和虚部构造复数	isreal	是否为实数组
	conj	复数的共轭	cplxpair	整理为共轭对
	imag	复数的虚部		
取整函数	fix	朝零方向取整	mod	模数（带符号余数）
	floor	朝负无穷-∞方向取整	rem	除后取余数
	ceil	朝正无穷+∞方向取整	sign	符号函数
	round	四舍五入到最近的整数		

2. 特殊函数

MATLAB 的特殊数学函数分为 3 部分：特殊数学函数、数理函数、坐标变换。表 2-5 中列出一些常用的函数，详细信息请查阅附录 A。

表 2-5　MATLAB 的特殊数学函数

函数名	功能	函数名称	功能
bessel	各类 Bessel 函数	rat	有理逼近
beta	β 函数	cross	矢量叉乘
gamma	γ 函数	dot	矢量点乘
ellipj	Jachobi 椭圆函数	cart2sph	直角坐标到球坐标变换
ellipke	完全椭圆积分	cart2pol	直角坐标到极坐标变换
erf	误差函数	pol2cart	极坐标变换到直角坐标
erfinv	逆误差函数	sph2cart	球坐标变换到直角坐标

2.5 矩 阵 分 解

2.5.1 特征值分解

如果有一矢量 v 和一常数λ，使得方阵 A 满足 $Av = \lambda v$，则λ称为特征值，而 v 称为特征矢量。

将特征值填入矩阵 Λ 的对角线位置上，并在其他位置上填入 0，再将特征矢量当成列矢量，合起来写成特征矢量矩阵 V，则有

$AV=V\Lambda$

如果 V 是一非奇异矩阵，则可以写出矩阵的特征值分解式

$A=V\Lambda V^{-1}$

矩阵的特征值分解调用函数 eig，其调用形式有以下几种：

- $[v, d]$ = eig(a)，得到矩阵 a 的特征值对角阵 d 和特征矢量矩阵 v，在这种情况下，矩阵的特征值分解为 $a\times v=v\times d$。
- $[v, d]$ = eig(a, 'nobalance')，为关闭平衡算法的形式。
- $[v, d]$ = eig(a, b)，得到矩阵 a 和 b 的广义特征值分解：$a\times v=b\times v\times d$。

【例 2-20】 矩阵 a 为$\begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\\7 & 8 & 9\end{bmatrix}$，矩阵 b 为$\begin{bmatrix}-2 & 1 & 3\\1 & 4 & -2\\2 & -1 & 2\end{bmatrix}$，计算 a 和 b 的广义特征值分解。

在命令窗口输入矩阵，计算$[v, d]$ = eig(a, b)：

```
>> a=[1 2 3;4 5 6;7 8 9];
>> b=[-2 1 3;1 4 -2;2 -1 2];
>> [v,d]=eig(a,b)
v =
    1.0000   -1.0000   -0.5000
    0.5499    0.5311    1.0000
    0.7110    0.2499   -0.5000
d =
    6.1970         0         0
    0    0.2474         0
    0         0   -0.0000
```

2.5.2 奇异值分解

如果存在两个矢量 u、v 及一常数 σ，使得矩阵 A 满足：

$Av=\sigma u$

$A'u=\sigma v$

则σ 称为奇异值，而 u、v 称为奇异矢量。

将奇异值写成对角方阵 Σ，而相对应的奇异矢量作为列矢量，则可写成两个正交矩阵

U、V，使得：

$AV = U\Sigma$

$A'U = V\Sigma$

因为 U、V 正交，所以可得奇异值表达式：

$A = U\Sigma V'$

对 m 行 n 列的矩阵 A 进行奇异值分解，可得 m 行 m 列的矩阵 U、m 行 n 列的矩阵 Σ 和 n 行 n 列的矩阵 V。

矩阵的奇异值分解由函数 svd 实现，其调用形式为：

- $[u, s, v]$ = svd(a) 或者$[u, s, v]$ = svd(a, 0)，使 $a == u \times s \times v'$。

对矩阵进行奇异值分解，矩阵的阶数没有限制，矩阵不必是方阵。

奇异值分解的应用很广，MATLAB 的算法中大量使用了奇异值分解，最典型的应用如最小二乘法和数据压缩等。

【例 2-21】 矩阵 a 为$[1\quad 1]$，对矩阵 a 进行奇异值分解。

在命令窗口输入矩阵，计算$[u, s, v]$ = svd(a)：

```
>> a=[1 1];
>> [u,s,v]=svd(a)
u =
    1
s =
    1.4142         0
v =
    0.7071   -0.7071
    0.7071    0.7071
```

2.5.3 LU 分解

LU 分解法是将方阵分解成一个下三角矩阵（或是以不同次序排列的下三角矩阵）和一个上三角矩阵，这类分解法又称为三角分解法，主要用于简化大矩阵的行列式值的计算过程、求逆矩阵和求解联立方程组。需要注意的是，这种分解法所得到的上、下三角阵并不是唯一的，可以找到多个不同的上、下三角阵对，每对三角阵相乘都会得到原矩阵。

MATLAB 使用函数 lu 实现 LU 分解法，其调用形式为：

- $[l, u]$ = lu(a)，其中 l 代表下三角阵，u 代表上三角阵。

【例 2-22】 矩阵 a 为$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$，求 a 的 LU 分解。

在命令窗口输入矩阵，计算$[l, u]$ = lu(a)：

```
>> a=[1 2 3;4 5 6;7 8 9];
>> [l,u]=lu(a)
l =
    0.1429    1.0000         0
    0.5714    0.5000    1.0000
```

```
    1.0000         0         0
u =
    7.0000    8.0000    9.0000
         0    0.8571    1.7143
         0         0    0.0000
```

2.5.4 Cholesky 分解

如果 A 为 n 阶对称正定矩阵，则存在一个非奇异的下三角实矩阵 L，使 $A=LL^{\mathrm{T}}$。当限定 L 的对角元素为正时，这种分解是唯一的，称为 Cholesky 分解。在 MATLAB 中，Cholesky 分解由函数 chol 实现。

【例 2-23】 矩阵 a 为 $\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 3 & 6 & 10 \\ 1 & 4 & 10 & 20 \end{bmatrix}$，求 a 的 Cholesky 分解。

矩阵 a 为特殊矩阵，称为 4 阶 pascal 矩阵，可在命令窗口用函数创建，计算 $c=\text{chol}(a)$:

```
>> a=pascal(4)
a =
     1     1     1     1
     1     2     3     4
     1     3     6    10
     1     4    10    20
>> c=chol(a)
c =
     1     1     1     1
     0     1     2     3
     0     0     1     3
     0     0     0     1
```

2.5.5 QR 分解

实矩阵 A 可以写成 $A=QR$ 的形式，其中 Q 为正交阵，R 为上三角阵。在 MATLAB 中，QR 分解由函数 qr 实现，其调用形式为：

- $[q, r]=\text{qr}(a)$　其中 q 代表正规正交矩阵，r 代表三角形矩阵。此外，原矩阵 a 不必一定是方阵，如果矩阵 a 是 $m\times n$ 阶的，则矩阵 q 是 $m\times m$ 阶的，矩阵 r 是 $m\times n$ 阶的。

【例 2-24】 矩阵 a 为 $\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 3 & 6 & 10 \\ 1 & 4 & 10 & 20 \end{bmatrix}$，求 a 的 QR 分解。

矩阵 a 为 4 阶 pascal 矩阵，可在命令窗口用函数创建，计算 $[q, r]=\text{qr}(a)$:

```
>> a=pascal(4)
a =
```

```
     1     1     1     1
     1     2     3     4
     1     3     6    10
     1     4    10    20
>> [q,r]=qr(a)
q =
   -0.5000    0.6708    0.5000    0.2236
   -0.5000    0.2236   -0.5000   -0.6708
   -0.5000   -0.2236   -0.5000    0.6708
   -0.5000   -0.6708    0.5000   -0.2236
r =
   -2.0000   -5.0000  -10.0000  -17.5000
         0   -2.2361   -6.7082  -14.0872
         0         0    1.0000    3.5000
         0         0         0   -0.2236
```

2.6　矩阵的特殊操作

2.6.1　特殊矩阵及其创建

1. 空阵

在 MATLAB 的命令窗口中输入[]，就创建了一个空阵。空阵具有如下性质：

- 在 MATLAB 的工作空间中存在被赋值的空阵变量；
- 空阵中不包括任何元素，是 0×0 阶的矩阵；
- 空阵可以在 MATLAB 的运算中传递。

2. 全 0 阵

全 0 阵是全部由 0 组成的矩阵或数组，可以是 $n \times n$ 阶的方阵，或者是 $m \times n$ 阶的非方阵，也可以是 $m \times n \times p$...阶的多维数组，由函数 zeros(m, n, p, ...)创建。

3. 单位阵

单位阵是对角线元素为 1，其他元素为 0 的矩阵或数组，可以是 $n \times n$ 阶的方阵，或者是 $m \times n$ 阶的非方阵，由函数 eye(m, n)创建。

4. 全 1 阵

全 1 阵是全部由 1 组成的矩阵或数组，可以是 $n \times n$ 阶的方阵，或者是 $m \times n$ 阶的非方阵，也可以是 $m \times n \times p$...阶的多维数组，由函数 ones(m, n, p, ...)创建。

5. 随机阵

随机阵是全部由值的范围在 0 到 1 区间内的元素组成的矩阵或数组，可以是 $n \times n$ 阶的方阵，或者是 $m \times n$ 阶的非方阵，也可以是 $m \times n \times p$...阶的多维数组，由函数 rand(m, n, p, ...)创建。其中一种特殊的随机阵为正态分布的随机阵，由函数 randn(m, n, p, ...)创建。

2.6.2 其他特殊矩阵

MATLAB 中提供了一些数学上比较著名的特殊矩阵。常用的特殊矩阵函数见表 2-6。

表 2-6 特殊矩阵函数

函数名	特殊矩阵	函数名	特殊矩阵
compan	伴随矩阵函数	magic	魔方矩阵
gallery	Higham 测试阵	pascal	Pascal 矩阵
hadamard	Hadamard 矩阵	rosser	经典对称特征值测试矩阵
hankel	Hankel 矩阵	toeplitz	Toeplitz 矩阵
hilb	Hilbert 矩阵	vander	Vander 矩阵
invhilb	反 Hilbert 矩阵	wilkinson	Wilkison's 特征值测试矩阵

2.6.3 矩阵的特殊操作

1. 重新排列

MATLAB 可以实现矩阵或数组元素的重新排列，以实现矩阵和数组尺寸或维数的变化，如将一个一维矢量重新排列为二维矩阵或多维数组。根据 MATLAB 数组元素的排列顺序规则，重新排列的元素按照先排列、再排行、然后排列第三维、第四维...的顺序排列。MATLAB 实现矩阵重新排列的函数为 reshape，其调用形式为:

- c = reshape($a, m, n, p, \ldots$)　其中 m、n、p 等分别为新数组各维的阶数。

需要注意的是，新数组的各维阶数的乘积必须与原数组的各维阶数的乘积相同。

【例 2-25】 矩阵 a 为 $\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 3 & 6 & 10 \\ 1 & 4 & 10 & 20 \end{bmatrix}$，将其重新排列为 1×16 阶的一维矢量和 2×4×2 阶的三维数组。

矩阵 a 为 4 阶 pascal 矩阵，可在命令窗口用函数创建，分别计算 c = reshape(a, 1, 16)和 c = reshape(a, 2, 4, 2):

```
>> a=pascal(4)
a =
     1     1     1     1
     1     2     3     4
     1     3     6    10
     1     4    10    20
>> c=reshape(a,1,16)
c =
Columns 1 through 10
     1     1     1     1     1     2     3     4     1     3
Columns 11 through 16
```

```
    6    10     1     4    10    20
>> c=reshape(a,2,4,2)
c(:,:,1) =
    1     1     1     3
    1     1     2     4
c(:,:,2) =
    1     6     1    10
    3    10     4    20
```

2. 矩阵的翻转和旋转

矩阵可以实现左右翻转、上下翻转、旋转等操作，其函数见表 2-7。

表 2-7　矩阵的翻转和旋转

函数名	功能	函数名称	功能
fliplr	矩阵左右翻转	flipdim	矩阵的第 n 维翻转
flipud	矩阵上下翻转	rot90	矩阵逆时针旋转 90°

【例 2-26】 矩阵 a 为 $\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 3 & 6 & 10 \\ 1 & 4 & 10 & 20 \end{bmatrix}$，分别将其左右翻转、上下翻转和旋转。

矩阵 a 为 4 阶 pascal 矩阵，可在命令窗口用函数创建，分别计算 $c=\text{fliplr}(a)$、$c=\text{flipud}(a)$ 和 $c=\text{rot90}(a)$。

```
>> a=pascal(4)
a =
    1     1     1     1
    1     2     3     4
    1     3     6    10
    1     4    10    20
>> c=fliplr(a)
c =
    1     1     1     1
    4     3     2     1
   10     6     3     1
   20    10     4     1
>> c=flipud(a)
c =
    1     4    10    20
    1     3     6    10
    1     2     3     4
    1     1     1     1
>> c=rot90(a)
c =
```

```
  1     4    10    20
  1     3     6    10
  1     2     3     4
  1     1     1     1
```

3. 矩阵的抽取

矩阵的抽取函数包括 diag、tril 和 triu。

函数 diag 实现矩阵对角元素的抽取，其调用格式为：

• $c = \mathrm{diag}(a, n)$ c 为抽取矩阵 a 的第 n 条对角线所创建的元素矢量，n 大于 0 时抽取矩阵上方的第 n 条对角线，n 小于 0 时抽取矩阵下方的第 n 条对角线，n 等于 0 或不指定 n 时抽取主对角线。

函数 diag 还可以逆向操作，用矢量创建对角矩阵，其调用格式为：

• $a = \mathrm{diag}(c, n)$ 创建对角矩阵 a，使矢量 c 成为 a 的第 n 条对角线元素，当 $n = 0$ 或不指定 n 时，c 为 a 的主对角线。

函数 *tril* 实现下三角矩阵抽取，其调用格式为：

• $c = \mathrm{tril}(a, n)$ 抽取矩阵 a 的第 n 条对角线下面的部分（包括第 n 条对角线），n 的定义同 diag 函数。

函数 *triu* 实现上三角矩阵抽取，其调用格式为：

• $c = \mathrm{triu}(a, n)$ 抽取矩阵 a 的第 n 条对角线上面的部分（包括第 n 条对角线），n 的定义同 diag 函数。

【例 2-27】 矩阵 a 为 $\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 3 & 6 & 10 \\ 1 & 4 & 10 & 20 \end{bmatrix}$，分别抽取其对角线元素、创建对角矩阵、抽取上三角矩阵和下三角矩阵。

矩阵 a 为 4 阶 pascal 矩阵，可在命令窗口用函数创建，分别计算 $c = \mathrm{diag}(a,1)$、$b = \mathrm{diag}(c,1)$、$c = \mathrm{tril}(a)$和 $c = \mathrm{triu}(a, -1)$。

```
>> a=pascal(4)
a =
     1     1     1     1
     1     2     3     4
     1     3     6    10
     1     4    10    20
>> c=diag(a,1)
c =
     1
     3
     10
>> b=diag(c,1)
b =
     0     1     0     0
     0     0     3     0
```

```
     0     0     0    10
     0     0     0     0
>> c=tril(a)
c =
     1     0     0     0
     1     2     0     0
     1     3     6     0
     1     4    10    20
>> c=triu(a,-1)
c =
     1     1     1     1
     1     2     3     4
     0     3     6    10
     0     0    10    20
```

习　题　2

2.1　在 MATLAB 中建立矩阵$\begin{bmatrix}5 & 7 & 3\\4 & 9 & 1\end{bmatrix}$，并将其赋予变量 a。

2.2　建立矩阵的方法有几种？各有什么优点？

2.3　进行算术运算时，数组运算和矩阵运算各有什么要求？

2.4　数组运算和矩阵运算的运算符有什么区别？

2.5　计算矩阵$\begin{bmatrix}5 & 3 & 5\\3 & 7 & 4\\7 & 9 & 8\end{bmatrix}$与$\begin{bmatrix}2 & 4 & 2\\6 & 7 & 9\\8 & 3 & 6\end{bmatrix}$之和。

2.6　求 $x=\begin{bmatrix}4+8i & 3+5i & 2-7i & 1+4i & 7-5i\\3+2i & 7-6i & 9+4i & 3-9i & 4+4i\end{bmatrix}$的共轭转置。

2.7　计算 $a=\begin{bmatrix}6 & 9 & 3\\2 & 7 & 5\end{bmatrix}$与 $b=\begin{bmatrix}2 & 4 & 1\\4 & 6 & 8\end{bmatrix}$的数组乘积。

2.8　“左除”与“右除”有什么区别？

2.9　对于 $AX=B$，如果 $A=\begin{bmatrix}4 & 9 & 2\\7 & 6 & 4\\3 & 5 & 7\end{bmatrix}$，$B=\begin{bmatrix}37\\26\\28\end{bmatrix}$，求解 X。

2.10　已知：$a=\begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\\7 & 8 & 9\end{bmatrix}$，分别计算 a 的数组平方和矩阵平方，并比较其结果。

2.11　$a=\begin{bmatrix}1 & 2 & 5\\3 & 6 & -4\end{bmatrix}$，$b=\begin{bmatrix}8 & -7 & 4\\3 & 6 & 2\end{bmatrix}$，观察 a 与 b 之间的 6 种关系运算的结果。

2.12　$a=[5\quad 0.2\quad 0\quad -8\quad -0.7]$，在进行逻辑运算时，$a$ 相当于什么样的逻辑量？

2.13　在 $\sin x$ 运算中，x 是角度还是弧度？

2.14　$x=[30\quad 45\quad 60]$，x 的单位为角度，求 x 的正弦、余弦、正切和余切。

2.15 用四舍五入的方法将数组[2.4568 6.3982 3.9375 8.5042]取整。

2.16 矩阵 $a=\begin{bmatrix}9 & 1 & 2\\5 & 6 & 3\\8 & 2 & 7\end{bmatrix}$，分别对 a 进行特征值分解、奇异值分解、LU 分解、QR 分解及 Cholesky 分解。

2.17 将矩阵 $a=\begin{bmatrix}4 & 2\\5 & 7\end{bmatrix}$、$b=\begin{bmatrix}7 & 1\\8 & 3\end{bmatrix}$ 和 $c=\begin{bmatrix}5 & 9\\6 & 2\end{bmatrix}$ 组合成两个新矩阵。

(1) 组合成一个 4×3 的矩阵，第一列为按列顺序排列的 a 矩阵元素，第二列为按列顺序排列的 b 矩阵元素，第三列为按列顺序排列的 c 矩阵元素，即

$$\begin{bmatrix}4 & 7 & 5\\5 & 8 & 6\\2 & 1 & 9\\7 & 3 & 2\end{bmatrix}$$

(2) 按照 a、b、c 的列顺序组合成一个行矢量，即 $\begin{bmatrix}4 & 5 & 2 & 7 & 7 & 8 & 1 & 3 & 5 & 6 & 9 & 2\end{bmatrix}$。

第 3 章 数值运算基础

3.1 多 项 式

3.1.1 创建多项式

对于多项式 $P(x)=a_0x^n+a_1x^{n-1}+\cdots+a_{n-1}x+a$，可以用它的系数矢量表示：

$$P=\begin{bmatrix}a_0 & a_1 \cdots a_{n-1} & a_n\end{bmatrix}$$

这样，在 MATLAB 中，将多项式问题转化为矢量问题。

创建多项式的方法有以下几种。

1. 系数矢量的直接输入法

在 MATLAB 的命令窗口直接输入多项式的系数矢量，然后利用转换函数 poly2sym 将多项式由系数矢量形式转换为符号形式。

【例 3-1】 输入系数矢量，创建多项式 x^3-4x^2+3x+2。

在命令窗口输入系数矢量，并转换为多项式：

```
>> poly2sym([1 -4 3 2])
ans =
      x^3-4*x^2+3*x+2
```

2. 特征多项式输入法

创建多项式的另一种方法是由矩阵的特征多项式取得，由函数 poly 实现。

需要说明的是，n 阶方阵的特征多项式系数矢量一定是 n+1 阶的，同时特征多项式系数矢量的第一个元素必须为 1。

【例 3-2】 求矩阵 $\begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\\7 & 8 & 0\end{bmatrix}$ 的特征多项式系数，并转换为多项式形式。

在命令窗口输入矩阵，求特征多项式系数并转换为多项式形式：

```
>> a=[1 2 3;4 5 6;7 8 0];
>> p=poly(a)
p =
    1.0000   -6.0000  -72.0000  -27.0000
>> poly2sym(p)
ans =
    x^3-6*x^2-72*x-27
```

3. 由根矢量创建多项式

由给定的根矢量也可以创建多项式，同样由函数 poly 实现。

由给定的根矢量创建多项式应该注意两点：

(1) 如果希望创建实系数多项式，根矢量的复数根必须共轭成对。

(2) 含复数根的根矢量所创建的多项式系数矢量的系数中，有可能带有很小（在截断误差数量级）的虚部，此时可采用取实部的命令（real）把虚部滤掉。

【例 3-3】 根据根矢量$[-0.5 \quad -0.3+0.4i \quad -0.3-0.4i]$创建多项式。

在命令窗口输入根矢量，并创建多项式：

```
>> r=[-0.5 -0.3+0.4i -0.3-0.4i];
>> p=poly(r)
p =
    1.0000    1.1000    0.5500    0.1250
>> pr=real(p)
pr =
    1.0000    1.1000    0.5500    0.1250
>> ppr=poly2sym(pr)
ppr =
    x^3+11/10*x^2+11/20*x+1/8
```

3.1.2 多项式运算

1. 求多项式的值

求多项式的值有两种算法，一种按数组运算规则计算，对应的函数为 polyval；另一种按矩阵的运算规则计算，对应的函数为 polyvalm。

函数 polyval 的调用格式为：

• y = polyval(p, x)　求多项式 p 在 x 点的值，x 也可以是一数组，表示求多项式 p 在各点的值。

• y = polyval(p, x, [], mu)　在这种用法中，用 $\hat{x}=\dfrac{x-\mu_1}{\mu_2}$ 代替 x，求多项式 p 在 x 点的值，其中 $mu=[\mu_1 \quad \mu_2]$，μ_1 = mean(x)，μ_2 = std (x)。

• [y, $delta$] = polyval(p, x, S) 和[y, $delta$] = polyval(p, x, S, mu)　使用可选的结构 S 产生误差估计 $y\pm delta$，其他用法同上。

【例 3-4】 求多项式$3x^2+2x+1$在 5、7 和 9 处的值。

在命令窗口输入多项式系数，并求值：

```
>> p = [3 2 1];
>> polyval(p,[5 7 9])
ans =
    86   162   262
```

函数 polyvalm 的调用格式为：

• y = polyvalm(p, x)　求多项式 p 对于矩阵 x 的值，要求矩阵 x 必须是方阵，x 如果是一标量，求得的值与函数 polyval 相同。

【例 3-5】 求多项式$3x^2+2x+1$对于矩阵$\begin{bmatrix}2 & 5\\7 & 9\end{bmatrix}$及标量 5 的值。

在命令窗口输入多项式系数，并求值：

```
>> p = [3 2 1];
```

```
>> polyvalm(p,[2 5;7 9])
ans =
   122   175
   245   367
>> polyvalm(p,5)
ans =
   86
```

2. 求多项式的根

MATLAB 进行多项式的求根运算时，可以有两种方法，一种是直接调用求根函数 roots；另一种是先把多项式转化为伴随矩阵，然后再求其特征值。在 MATLAB 中约定，多项式系数用行矢量表示，一组根用列矢量表示。

【例 3-6】　分别用两种方法求多项式 $x^5-5x^4+3x^3-6x^2+4x-10$ 的根。

在命令窗口输入多项式系数，并求根：

```
>> a=[1 -5 3 -6 4 -10];
>> r=roots(a)
r =
    4.6130
    0.7621 + 0.9789i
    0.7621 - 0.9789i
   -0.5685 + 1.0419i
   -0.5685 - 1.0419i
>> s=compan(a)
s =
     5    -3     6    -4    10
     1     0     0     0     0
     0     1     0     0     0
     0     0     1     0     0
     0     0     0     1     0
>> r=eig(s)
r =
    4.6130
    0.7621 + 0.9789i
    0.7621 - 0.9789i
   -0.5685 + 1.0419i
   -0.5685 - 1.0419i
```

可以看出，用两种方法求得的根是相同的。

3. 多项式的乘除运算

多项式的乘法由函数 conv 实现，conv 也是矢量的卷积函数；多项式的除法由函数 deconv 实现，deconv 也是矢量的卷积函数的逆函数。

【例 3-7】　计算两多项式 $x^4-5x^3+3x^2-4x+2$ 和 x^3+2x^2-5x+3 的乘法。

在命令窗口输入多项式系数，求两多项式的乘法：

```
>> a=[1 -5 3 -4 2];
```

```
>> b=[1 2 -5 3];
>> c=conv(a,b)
c =
    1    -3   -12    30   -36    33   -22     6
```

【例 3-8】 计算例 3-7 中求得的乘积被 x^3+2x^2-5x+3 除所得结果。

利用例 3-7 中求得的结果，求两多项式的除法：

```
>> d=deconv(c,b)
d =
    1    -5     3    -4     2
```

4. 多项式的微积分

多项式的微分由函数 polyder 实现，多项式的积分由函数 polyint 实现。

【例 3-9】 计算多项式 $3x^4-5x^3+2x^2-6x+10$ 的微分。

在命令窗口输入多项式系数，求多项式的微分：

```
>> p=[3 -5 2 -6 10];
>> polyder(p)
ans =
    12   -15     4    -6
>> poly2sym(ans)
ans =
    12*x^3-15*x^2+4*x-6
```

【例 3-10】 计算多项式 $12x^3-15x^2+4x-6$ 的积分。

在命令窗口输入多项式系数，求多项式的积分：

```
>> p=[12 -15 4 -6];
>> polyint(p)
ans =
    3    -5     2    -6     0
```

5. 多项式的部分分式展开

对于多项式 $b(x)$和不含重根的 n 阶多项式 $a(x)$之比，有如下的展开：

$$\frac{b(x)}{a(x)}=\frac{r_1}{x-p_1}+\frac{r_2}{x-p_2}+\cdots+\frac{r_n}{x-p_n}+k(x)$$

式中 p_1，p_2，…，p_n称为极点（Poles），r_1，r_2，…，r_n称为留数（Residues），$k(x)$称为直项（Direct term）。假如 $a(x)$有 m 重根 $p(j)=...=p(j+m-1)$，则相应的部分写成：

$$\frac{r_j}{x-p_j}+\frac{r_{j+1}}{(x-p_j)^2}+\cdots+\frac{r_{j+m-1}}{(x-p_j)^m}$$

在 MATLAB 中，计算部分分式展开的函数为 residue，其调用格式为：

- $[r, p, k]$ = residue(b, a) 计算由多项式表达式 $b(x)$和 $a(x)$分别为分子和分母时所得到的留数 r、极点 p 和直项 k。
- $[b, a]$ = residue(r, p, k) 将部分分式展开转换回多项式表达式 $b(x)$和 $a(x)$的系数。

【例 3-11】 两多项式的比为 $\dfrac{b(x)}{a(x)}=\dfrac{5x^3+3x^2-2x+7}{-4x^3+8x+3}$，求部分分式展开。

在命令窗口输入多项式系数，求部分分式展开：

```
>> a = [-4 0 8 3];
>> b = [ 5 3 -2 7];
>> [r, p, k] = residue(b,a)
r =
    -1.4167
    -0.6653
    1.3320
p =
    1.5737
    -1.1644
    -0.4093
k =
    -1.2500
```

【例 3-12】 利用在例 3-11 中求得的部分分式展开结果转换回多项式表达式 $b(x)$和 $a(x)$的系数。

利用在例 3-11 中求得的结果，求多项式系数：

```
>> [b,a] = residue(r,p,k)
b =
    -1.2500   -0.7500    0.5000   -1.7500
a =
    1.0000   -0.0000   -2.0000   -0.7500
```

这里要注意的是，得到的结果为 $\dfrac{b(x)}{a(x)}=\dfrac{-1.25x^3-0.75x^2+0.5x-1.75}{x^3-2x-0.75}$，这是对于表达式分母的归一形式。

6. 多项式拟合

多项式拟合的函数为 polyfit，其调用格式为：

- p = polyfit(x, y, n)　应用最小二乘法求出 n 阶拟合多项式 $p(x)$，即用 $p(x(i))$拟合 $y(i)$。
- [p, S] = polyfit(x, y, n)　返回 n 阶拟合多项式 $p(x)$和包括误差估计的结构 S。
- [p, S, mu] = polyfit(x, y, n)　在这种用法中，用 $\hat{x}=\dfrac{x-\mu_1}{\mu_2}$ 代替 x，求拟合多项式 p，其中 $mu=[\mu_1\ \ \mu_2]$，μ_1= mean(x)，μ_2= std (x)。

【例 3-13】 求误差函数的 6 阶拟合多项式。

在命令窗口产生自变量 x，作出误差函数，然后求 6 阶拟合多项式（图 3-1）：

```
>> x = (0: 0.1: 2.5)';                %生成 0 至 2.5 间隔为 0.1 的自变量
>> y = erf(x);                        %计算误差函数
>> p = polyfit(x,y,6)                 %求 6 阶拟合多项式
p =
    0.0084   -0.0983    0.4217   -0.7435    0.1471    1.1064    0.0004
>> x = (0: 0.1: 5)';                  %生成 0 至 5 间隔为 0.1 的自变量
>> y = erf(x);                        %计算误差函数
```

```
>> f = polyval(p,x);                    %计算拟合函数的值
>> plot(x,y,'o',x,f,'-')                %绘图函数，其用法将在 5.1 中介绍
>> axis([0 5 0 2])
```

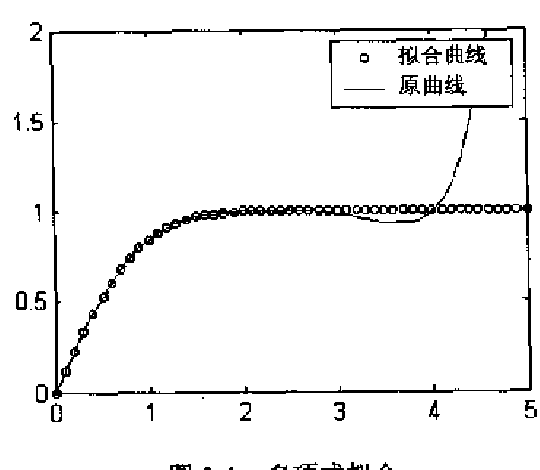


图 3-1 多项式拟合

多项式 $0.0084x^6 - 0.0983x^5 + 0.4217x^4 - 0.7435x^3 + 0.1471x^2 + 1.1064x + 0.0004$ 为拟合所得。由于拟合是在[0, 2.5]区间进行的，从图 3-1 中可以看出，在此区间内，拟合曲线非常接近原曲线，而在区间外，两曲线的差别很大。

3.2 线性代数

在分析及解决问题的过程中，通常根据已知条件尝试将系统以方程的形式来表示，再由求出的方程解来进一步了解系统，所以解线性方程是非常重要的。

解线性方程就是找出是否存在一个唯一矩阵 x，使得矩阵 a、b 满足以下关系：

$$ax = b \qquad 或 \qquad xa = b$$

MATLAB 以斜线和反斜线来表示除法运算因子，其中：

$x = a\backslash b$ 是方程式 $ax=b$ 的解。

$x = b/a$ 是方程式 $xa=b$ 的解。

不论是 $x = a\backslash b$ 还是 $x = b/a$，系数矩阵 a 都是分母。此外，在 $x = a\backslash b$ 中，矩阵 a 与 b 的行数相同，而解 x 的列数与矩阵 b 的列数相同，行数则等于矩阵 a 的列数。同样，$x = b/a$ 中矩阵 x 和矩阵 a、b 的行列关系恰好和 $x = a\backslash b$ 相反。

通常线性方程大多数写成 $ax = b$，很少写成 $xa = b$，所以反斜线“\”较常使用，而斜线“/”则很少使用，两者间有下面的置换关系：

$$(b/a)' = (a'\backslash b')$$

系数矩阵 a 并不一定是方阵，所以针对 m 行 n 列的系数矩阵，可能有以下 3 种情况：

(1) $m = n$（方阵系统），可以尝试计算精确解。

(2) $m > n$（超定系统），可以尝试计算最小二乘解。

(3) $m < n$（欠定系统），可以尝试计算含有最少 m 的基解。

反斜线运算因子对于不同形式的参数矩阵，采用不同的运算法则来处理。MATLAB 会自动检测参数矩阵，以区分下面几种形式：

- 三角矩阵
- 对称正定矩阵
- 非奇异方阵
- 超定系统
- 欠定系统

3.2.1 方阵系统

最常见的线性方程是系数矩阵为方阵 a 和由常数项组成列矢量 b 的情况，则解 x 可写成 $x=a\backslash b$，其中 x 和 b 的尺寸相同。

【例 3-14】 求方阵系统的根。

在命令窗口产生随机方阵 a 和矢量 b，计算 $a\backslash b$：

```
>> a=fix(15*rand(3,3))
a =
    14     7     6
     3    13     0
     9    11    12
>> b=fix(15*rand(3,1))
b =
     6
     9
    11
>> x=a\b
x =
   -0.0588
    0.7059
    0.3137
```

【例 3-15】 a 和 b 均为方阵，求方阵系统的根。

在命令窗口产生随机方阵 a 和 b，计算 $a\backslash b$：

```
>> a=fix(15*rand(3,3))
a =
    13     6     6
    11    14    13
     2    13     0
>> b=fix(15*rand(3,3))
b =
     5     2     9
    12     3     4
     0     2     2
>> x=a\b
x =
   -0.0674    0.0859    0.9038
```

```
    0.0104    0.1406    0.0148
    0.9689    0.0067   -0.4730
```

假如方阵 a 的各个行矢量线性相关，则称方阵 a 为奇异矩阵，这也使线性方程 $ax=b$ 有无穷多组解。

假如方阵 a 近似奇异矩阵，则反斜线运算因子将发出警告信息。假如方阵 a 确定是奇异矩阵，则反斜线运算因子将给出错误信息。

3.2.2 超定系统

线性方程的超定系统是指方程的个数多于自变量的个数的系统。求解超定系统一般采用最小二乘法。

假如有一组实验数据，在时间 t 测得数值 y，得到的数据如表 3-1 所示。

表 3-1 一组测试数据

t	y
0.0	0.82
0.3	0.72
0.8	0.63
1.1	0.60
1.6	0.55
2.2	0.50

这些数据被认为具有下面的衰减指数趋势：

$$y(t) \sim c_1 + c_2 e^{-t}$$

这个方程指出矢量 y 可以由两个矢量逐步逼近而得，这两个矢量一个是单行的常数矢量，另一个是由指数 e^{-t} 项构成的矢量。至于两个参数 c_1、c_2，则可通过最小二乘法求得，它们表示实验数据到方程 $y(t) \sim c_1 + c_2 e^{-t}$ 之间距离的最小平方和。

将实验数据代入方程式 $y(t) \sim c_1 + c_2 e^{-t}$ 中，可以得到含有两个未知数的 6 个等式，可以将其写成 6 行 2 列的矩阵 e，如例 3-16。

【例 3-16】 求表 3-1 数据的最小二乘解。

在命令窗口输入数据 t 和 y，产生矩阵 e，然后计算 $e\backslash y$：

```
>> t=[0 0.3 0.8 1.1 1.6 2.3]';
>> y=[0.82 0.72 0.63 0.60 0.55 0.50]';
>> e=[ones(size(t)) exp(-t)]
e =
    1.0000    1.0000
    1.0000    0.7408
    1.0000    0.4493
    1.0000    0.3329
    1.0000    0.2019
    1.0000    0.1003
>> c=e\y
```

```
c =
    0.4760
    0.3413
>> t1=[0:0.1:2.5]';
>> y1=[ones(size(t1)),exp(-t1)]*c;
>> plot(t1,y1,'k',t,y,'ro')
```

将求得的解代入方程式，可得：

$$y(t) \sim 0.4760 + 0.3413\mathrm{e}^{-t}$$

最后的曲线如图 3-2 所示。

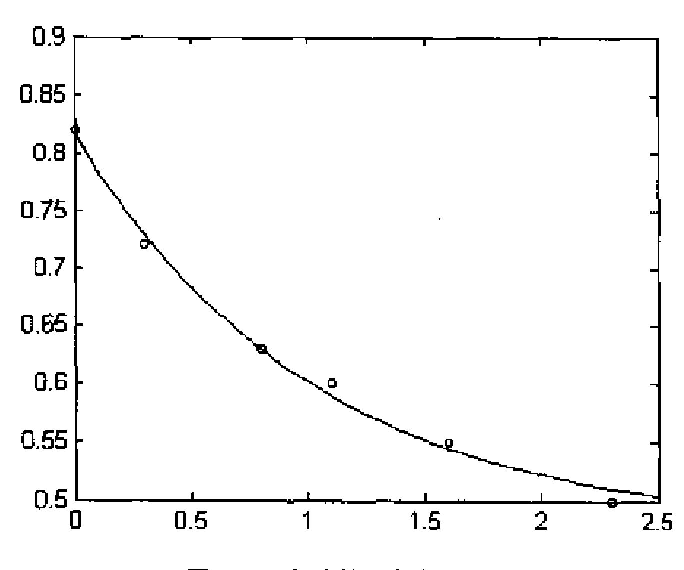


图 3-2　实验数据与拟合曲线

如果一个矩阵的行矢量是线性相关的，则 $ax=b$ 的最小二乘解并不唯一，因此 $a\backslash b$ 运算将给出警告，并产生含有最少元素的基解。

3.2.3　欠定系统

如果一个系统中未知数的个数比方程式的个数多，称系统为欠定系统，或称线性相关系统。在线性领域里，这种系统还会伴随约束条件，但这里不讨论有约束条件的情况，反斜线运算因子也只讨论针对无约束条件的情况。

欠定系统的解都不唯一，MATLAB 会计算一组构成通解的基解，而方程的特解则由 QR 分解法来决定。

欠定系统有两种算法，即最少元素解 $a\backslash b$ 和最小范数解 pinv(a)*b。

【例 3-17】　求解欠定系统。

在命令窗口产生随机矩阵 a 和 b，并计算 $a\backslash b$：

```
>> a=fix(15*rand(3,3))
a =
     0    13    12
    11     6     7
     6     6     3
```

```
>> b=fix(15*rand(3,1))
b =
    10
    12
     0
>> p=a\b
p =
     0.3270
    -1.6226
     2.5912
```

【例 3-18】 使用两种方法求解欠定系统。

在命令窗口创建矩阵 a 和矢量 b，分别计算 $a\backslash b$ 和 pinv(a)*b：

```
>> a=[1 1 1;1 1 -1]
>> b=[10 6]'
>> p=a\b
>> q=pinv(a)*b
a =
    1     1     1
    1     1    -1
b =
    10
     6
p =
    8.0000
         0
    2.0000
q =
    4.0000
    4.0000
    2.0000
```

3.3 数据分析

MATLAB 对数据分析命令有两条约定：

(1) 若输入量 X 为矢量，则不论是行矢量还是列矢量，运算是对整个矢量进行的。

(2) 若输入量 X 为矩阵，则函数运算按列进行，即默认每列是由一个变量的不同"观察"所得的数据组成。

这两条约定不仅适用于 MATLAB 的数据分析函数，而且也适用于所有的 MATLAB 函数。因此建议用户编制程序时，尽量遵循以上约定，以便更好地利用 MATLAB 现有的函数和命令。

3.3.1　基本统计命令

MATLAB 提供了一系列数据统计函数，通过运用这些函数，可以完成基本的数据统计分析。如果进行深入的数理统计和概率分析，可以借助于 MATLAB 的统计工具箱（Statistic Toolbox），该工具箱提供了 200 多个函数，可以解决各种复杂的数理统计和概率分析问题。由于篇幅的限制，本节只介绍 MATLAB 基本平台中的统计功能，表 3-2 列出了常用的统计函数。

表 3-2　基本统计函数

函数	功能
max(x)	求 x 各列的最大元素
mean(x)	求 x 各列的平均值
median(x)	求 x 各列的中位元素
min(x)	求 x 各列的最小元素
std(x)	求 x 各列的标准差
prod(x)	求 x 各列元素之积
sum(x)	求 x 各列元素之和
S=cumsum(x)	求 x 各列元素累计和（可用于积分计算） S 与 x 同阶，且 $S_{ij}=\sum_{k=1}^{i} x_{kj}$
P=cumprod(x)	求 x 各列元素累计积 P 与 x 同阶，且 $P_{ij}=\prod_{k=1}^{i} x_{kj}$
sort(x)	使 x 的各列元素按递增排序

【例 3-19】　基本统计函数的应用。

在命令窗口产生 100×4 阶的随机数组 A，计算 A 的最大值矢量、中位值矢量、平均值矢量和标准差矢量：

```
>> A=randn(100,4);
>> Amax=max(A)
Amax =
    2.1832    2.1122    2.3093    2.3726
>> Amed=median(A)
Amed =
    0.1162   -0.2074   -0.0258   -0.1141
>> Amean=mean(A)
Amean =
    0.0479   -0.1270   -0.0782   -0.0099
>> Astd=std(A)
Astd =
    0.8685    0.9447    0.9569    0.9977
```

MATLAB 中规定：在进行统计运算时，如果变量不是矢量而是矩阵，则按列进行运算，运算结果不是一个数值而是一个行矢量。以求最大值为例，max(A)的结果是一个行矢量，代表了每一列的最大值，如果希望求得 A 的最大值，可以再计算一次，如 $Amax$ = max(max(A))，也可以用 $Amax$ = max(A(:))。

3.3.2 协方差阵和相关阵

在 MATLAB 中，函数 conv 用于求矩阵的协方差矩阵，如果一个矩阵为 m 行 n 列，则其协方差矩阵为 n 阶方阵。

函数 corrcoef 用于求矩阵的相关系数矩阵，相关系数矩阵的元素排列与协方差矩阵的元素排列相同。相关系数是衡量两个矢量线性关系密切程度的量，如果两个矢量组相等则其相关系数为 1，如果两个矢量组相互独立则其相关系数为 0。表 3-3 中列出了协方差函数 cov 和相关系数函数 corrcoef 的算法。

表 3-3 函数 cov 和 corrcoef 的算法

函数	功能
C=cov(x)	求协方差阵 $\mathrm{cov}\|x\| = \mathrm{E}\|(x-\mu_x)^T(x-\mu_x)\|$
C=cov(x,y)	求两随机变量的协方差 $\begin{bmatrix} \sigma_x^2 & \mathrm{cov}(x,y) \\ \mathrm{cov}(y,x) & \sigma_y^2 \end{bmatrix}$
P=corrcoef(x)	求相关阵，且 $P(i,j)=\dfrac{C(i,j)}{\sqrt{C(i,i)}\sqrt{C(j,j)}}$
P=corrcoef(x,y)	求两随机变量的（2×2）相关系数

【例 3-20】 计算协方差和相关系数矩阵。

在命令窗口产生两个 10×3 阶的随机数组 x 和 y，计算关于 x 和 y 的协方差和相关系数矩阵：

```
>> x=rand(10,3);
>> y=rand(10,3);
>> cx=cov(x)
cx =
     0.0893   -0.0586   -0.0320
    -0.0586    0.0719    0.0298
    -0.0320    0.0298    0.0617
>> cy=cov(y)
cy =
     0.0805   -0.0308    0.0099
    -0.0308    0.0761   -0.0548
     0.0099   -0.0548    0.0667
>> cxy=cov(x,y)
cxy =
```

```
    0.0978   -0.0211
   -0.0211    0.0696
>> px=corrcoef(x)
px =
    1.0000   -0.7309   -0.4309
   -0.7309    1.0000    0.4479
   -0.4309    0.4479    1.0000
>> pxy= corrcoef(x,y)
pxy =
    1.0000   -0.2561
   -0.2561    1.0000
```

3.3.3 微分、差分与梯度

微分、差分和梯度运算是重要的数学运算方法，使用 MATLAB 提供的函数，可以解决大多数微分、差分及梯度计算的问题。

1. 微分及差分

求微分、差分的函数为 diff，其调用格式为：

• *Y* = diff(*X*)　计算相邻元素的差分。对于矢量，该运算返回一个较原矢量长度少一个元素的矢量，其值为[*X*(2)−*X*(1)　*X*(3) −*X*(2)　...　*X*(*n*) −*X*(*n*−1)]；对于矩阵，该运算返回一个较原矩阵少一行的矩阵，其值为矩阵列的差分[*X*(2:*m*,:)−X(1:*m*−1,:)]；对于数组，该运算返回沿第一个非独立维的差分。

• *Y* = diff(*X*, *n*)　求 *n* 阶差分，即 diff(*X*,2)表示 *diff*(diff(*X*))。

• *Y* = diff(*X*, *n*, *dim*)　按指定的维数 *dim* 求 *n* 阶差分，如果差分阶次 *n* 大于或等于指定维数 *dim*，运算将返回空数组。

【例 3-21】 计算三维数组的差分。

在命令窗口产生一个 3×3×2 阶的随机数组 *a*，计算 *a* 的前 6 阶差分：

```
>> a=rand(3,3,2);
>> a
a(:,:,1) =
    0.5341    0.8385    0.7027
    0.7271    0.5681    0.5466
    0.3093    0.3704    0.4449
a(:,:,2) =
    0.6946    0.9568    0.1730
    0.6213    0.5226    0.9797
    0.7948    0.8801    0.2714
>> diff(a)
ans(:,:,1) =
     0.1930   -0.2704   -0.1562
    -0.4178   -0.1977   -0.1017
ans(:,:,2) =
```

```
   -0.0733   -0.4343    0.8068
    0.1735    0.3576   -0.7083
>> diff(a,2)
ans(:,:,1) =
   -0.6109    0.0728    0.0545
ans(:,:,2) =
    0.2468    0.7918   -1.5151
>> diff(a,3)
ans(:,:,1) =
    0.6836   -0.0183
ans(:,:,2) =
    0.5450   -2.3069
>> diff(a,4)
ans(:,:,1) =
   -0.7019
ans(:,:,2) =
   -2.8519
>> diff(a,5)
ans =
   -2.1500
>> diff(a,6)
ans =
   Empty matrix: 1-by-0
```

2. 近似梯度

求近似梯度的函数为 gradient，其调用格式为：

- FX = gradient(F) 返回 F 的一维数值梯度，$FX=\dfrac{\partial F}{\partial x}$。
- [FX, FY] = gradient(F) 返回 F 的二维数值梯度，$FX=\dfrac{\partial F}{\partial x}$，$FY=\dfrac{\partial F}{\partial y}$。
- [Fx, Fy, Fz, ...] = gradient(F) 返回 F 的 n 维数值梯度。
- [...] = gradient(F, $h1$, $h2$, ...) 指定各方向的点间隔，如果不指定，缺省值为 1。

【例 3-22】 计算表达式 $z=xe^{-x^2-y^2}$ 的梯度。

在命令窗口使用函数 meshgrid 产生数据网格，定义表达式 z，计算数值梯度，并绘制等高线图（图 3-3）：

```
>> v = -2:0.2:2;                      %生成-2 到 2 间隔为 0.2 的自变量
>> [x,y] = meshgrid(v);               %产生数据网格
>> z = x .* exp(-x.^2 - y.^2);        %计算 z
>> [px,py] = gradient(z,.2,.2);       %求二维梯度
>> contour(x,y,z)                     %绘制等高线
>> hold on                            %保持绘图
>> quiver(x,y,px,py)                  %绘制矢量图
>> hold off
```

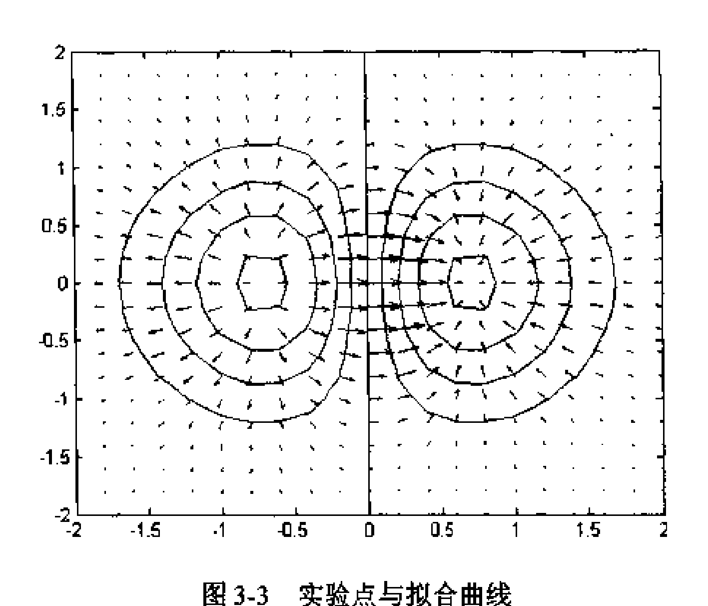


图 3-3　实验点与拟合曲线

3.4　插　　值

插值是在已知数据之间计算估计值的过程，在信号处理和图形分析中，插值运算有广泛的应用，MATLAB 提供了多种插值函数，以满足不同的需求。

3.4.1　一维插值

一维插值是最常用的插值运算，也是信号处理和曲线拟合等领域的基本运算。一维插值由函数 interp1 实现，该函数应用多项式技术计算插值点，其一般的调用格式为：

• *yi* = interp1(*x*, y, *xi*, *method*)　其中 y 为函数值矢量，x 为自变量的取值范围，x 与 y 的长度相同。x_i 为插值点的自变量矢量，*method* 为插值方法选项。对于一维插值，MATLAB 6.1 提供了 4 种方法：

(1) 邻近点插值（*method* = 'nearest'）　这种插值方法将插值结果的值设置为最近数据点的值。

(2) 线性插值（*method* = 'linear'）　这种插值方法在两个数据点之间连接直线，根据给定的插值点计算出它们在直线上的值，作为插值结果。该方法是 interp1 函数的缺省方法。

(3) 三次样条插值（*method* = 'spline'）　这种插值方法通过数据点拟合出三次样条曲线，根据给定的插值点计算出它们在曲线上的值，作为插值结果。

(4) 立方插值（*method* = 'pchip' 或 'cubic'）　这种插值方法通过分段立方 Hermite 插值方法计算插值结果。需要注意的是，MATLAB 6 所使用的立方插值方法与版本 5 不同，为了与版本 5 兼容，MATLAB 6 保留了版本 5 中的立方插值方法，如果使用该方法应选择(*method* = 'v5cubic')。

选择一种插值方法时，考虑的因素包括运算时间、占用计算机内存和插值的光滑程度。一般地说，插值的结果越光滑，所需的时间和内存的占用就越多，反之亦然。对于上述的 4 种插值方法，可以做相对的比较：

- 邻近点插值方法的速度最快，但平滑性最差；
- 线性插值方法占用的内存较邻近点插值方法多，运算时间也稍长，与邻近点插值不同，其结果是连续的，但在顶点处的斜率会改变；
- 三次样条插值方法的运算时间最长，但内存的占用较立方插值方法要少，在4种方法中，三次样条插值的平滑性最好，但如果输入数据不一致或数据点过近，可能出现很差的插值结果；
- 立方插值方法较邻近点插值和线性插值需要更多的内存和运算时间，其插值数据和导数都是连续的。

由于在很多情况下三次样条插值方法的插值效果最好，MATLAB 还专门提供了三次样条插值函数 spline。

【例 3-23】 一维插值函数插值方法的对比。

编辑命令函数，计算4种插值方法，并用图形显示（图 3-4）：

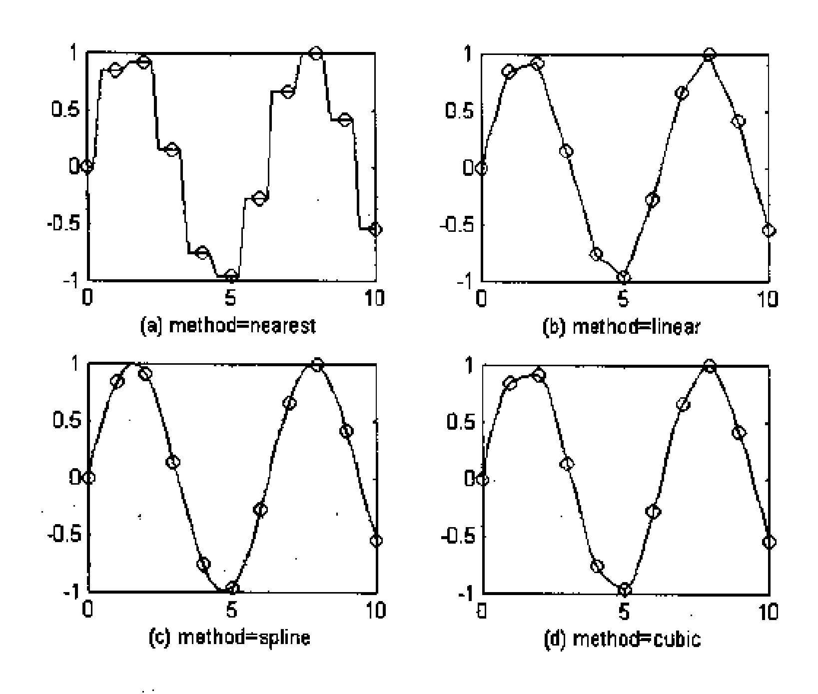


图 3-4 一维插值的 4 种方法

```
x=0:10;  y=sin(x);  xi=0:.25:10;
%将插值方法定义为单元数组
strmod={'nearest','linear','spline','cubic'}
%将 X 轴标识定义为单元数组
strlb={'(a) method=nearest', '(b) method=linear',...
       '(c) method=spline', '(d) method=cubic'};
for i=1:4
    yi=interp1(x,y,xi,strmod{i});
    %在一个图形窗口绘制多幅图形
    subplot(2,2,i),plot(x,y,'ro',xi,yi,'b'),xlabel(strlb(i))
end
```

【例 3-24】 三次样条插值。

在命令窗口输入命令，计算三次样条插值，并用图形显示（图 3-5）：

```
>> x0=0:10;  y0=sin(x0);
>> x=0:.25:10;
>> y=spline(x0,y0,x);
>> plot(x0,y0,'or',x,y,'k')
```

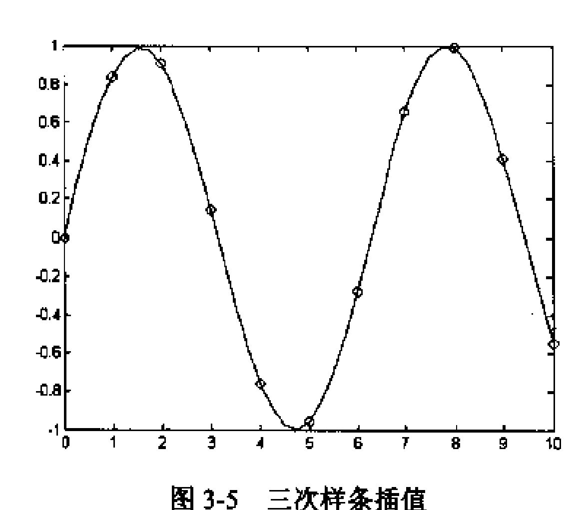


图 3-5　三次样条插值

从图 3-4 和图 3-5 可以看出，spline 函数计算的结果与 interp1 函数中使用 spline 方法所得到的结果是相同的。

3.4.2　二维插值

二维插值也是常用的插值运算方法，主要应用于图形图像处理和三维曲线拟合等领域。二维插值由函数 interp2 实现，其一般的调用格式为：

• *ZI* = interp2(*X*, *Y*, *Z*, *XI*, *YI*, *method*)　其中 *X* 和 *Y* 为由自变量组成的数组，*X* 与 *Y* 的尺寸相同，*Z* 为二维函数数组。*XI* 和 *YI* 为插值点的自变量数组，*method* 为插值方法选项。对于二维插值，MATLAB 6.1 也提供了 4 种方法：

(1) 邻近点插值（*method* = 'nearest'）。

(2) 双线性插值（*method* = 'linear'）。该方法是 interp2 函数的缺省方法。

(3) 三次样条插值（*method* = 'spline'）。

(4) 二重立方插值（*method* = 'cubic'）。

【例 3-25】　二维插值 4 种方法的对比。

在命令窗口输入命令，计算 4 种二维插值结果，并用图形显示（图 3-6、图 3-7）：

```
>> [x,y,z]=peaks(7);                          %生成双峰函数值
>> mesh(x,y,z)                                %绘制网格图
>> [xi,yi]=meshgrid(-3:0.2:3,-3:0.2:3);       %生成供插值的数据网格
>> z1=interp2(x,y,z,xi,yi,'nearest');
>> z2=interp2(x,y,z,xi,yi,'linear');
>> z3=interp2(x,y,z,xi,yi,'spline');
>> z4=interp2(x,y,z,xi,yi,'cubic');
>> mesh(xi,yi,z1)                             %绘制邻近点方法插值结果的网格图
```

```
>> mesh(xi,yi,z2)                    %绘制双线性方法插值结果的网格图
>> mesh(xi,yi,z3)                    %绘制三次样条方法插值结果网格图
>> mesh(xi,yi,z4)                    %绘制二重立方方法插值结果网格图
```

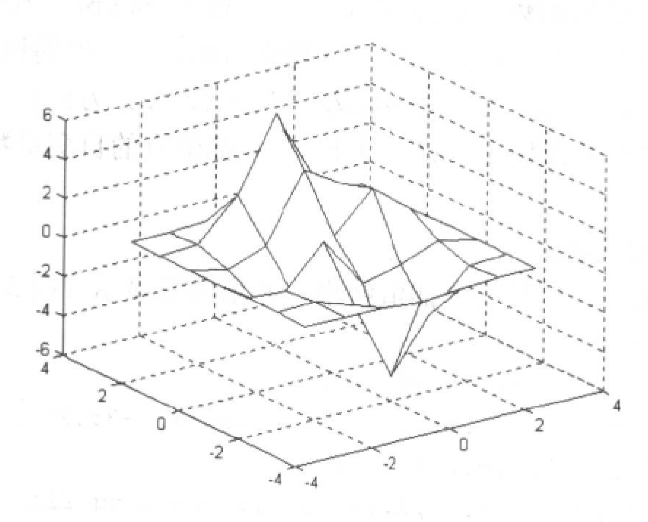
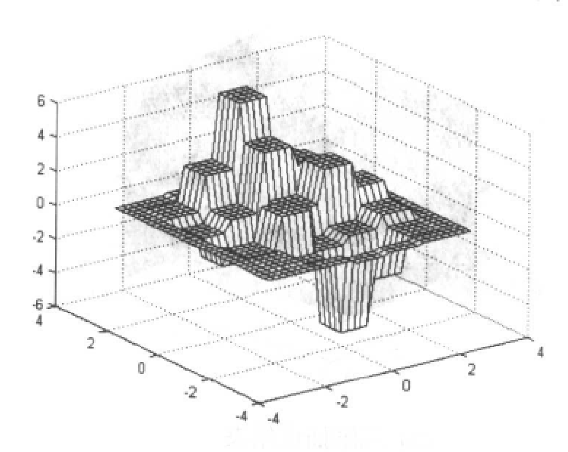
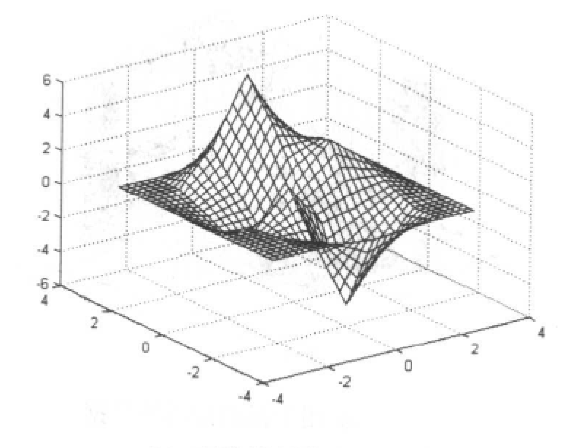


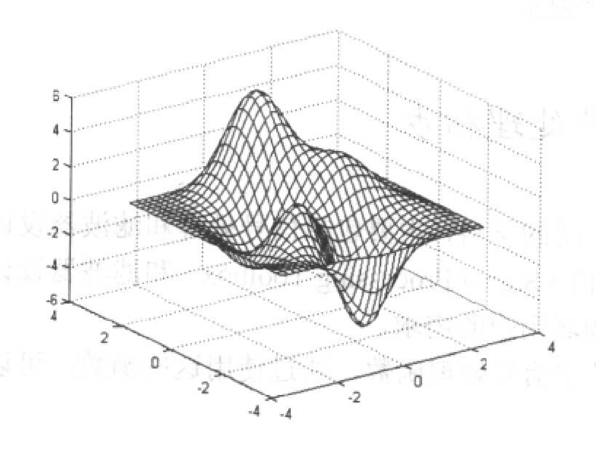
图 3-6 用于插值的原始数据



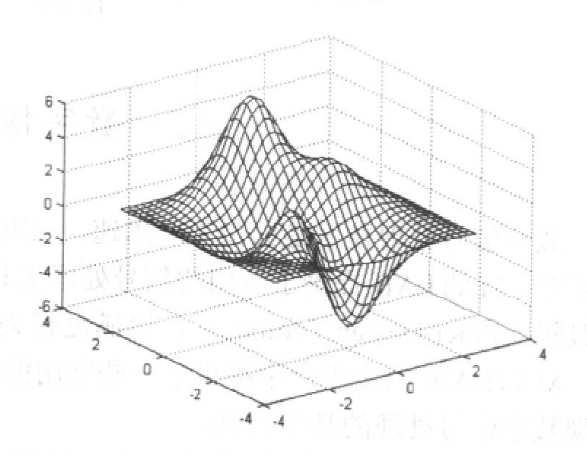
(a) 二维邻近点插值方法



(b) 双线性插值方法



(c) 三次样条插值方法



(d) 二重立方插值方法

图 3-7 4 种二维插值方法对比

3.4.3 多维插值

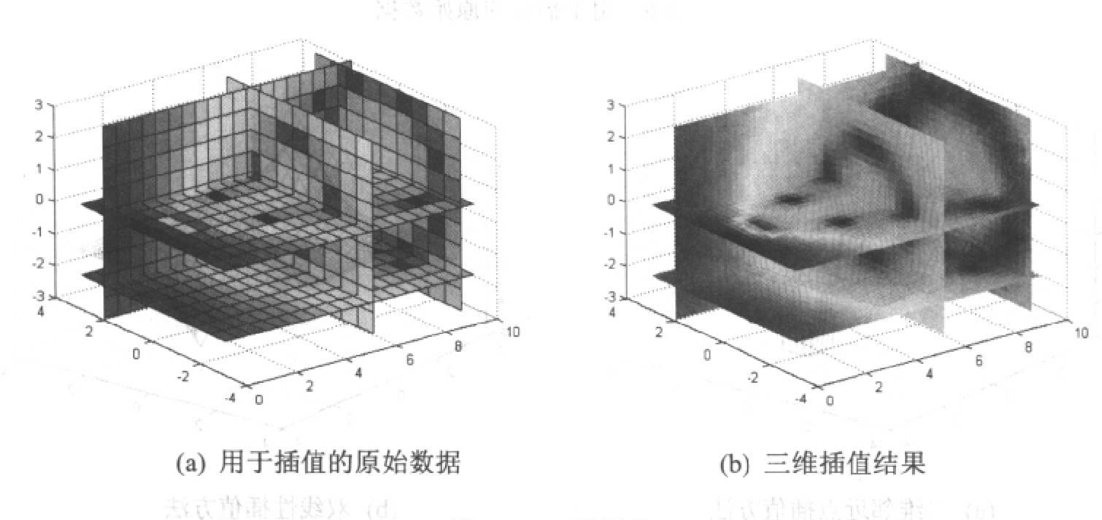
多维插值包括三维插值函数 interp3 和 n 维插值函数 interpn，其函数的调用方式及插值方法选项与一维、二维插值基本相同。以三维插值为例，其一般的调用格式为：

- *VI* = interp3(*X*, *Y*, *Z*, *V*, *XI*, *YI*, *ZI*, *method*)　其中 *X*、*Y*、*Z* 为由自变量组成的数组，*X*、*Y*、*Z* 的尺寸相同，*V* 为三维函数数组。*XI*、*YI* 和 *ZI* 为插值点的自变量数组，*method* 为插值方法选项。

【例 3-26】　三维插值实例。

在命令窗口输入命令，计算三维插值结果，并用图形显示（图 3-8）：

```
>> [x,y,z,v] = flow(10);
>> slice(x,y,z,v,[6 9.5],2,[-2 .2])
>> [xi,yi,zi] = meshgrid(.1:.25:10, -3:.25:3, -3:.25:3);
>> vi = interp3(x,y,z,v,xi,yi,zi);
>> slice(xi,yi,zi,vi,[6 9.5],2,[-2 .2]); shading flat
```



(a) 用于插值的原始数据　　(b) 三维插值结果

图 3-8　三维插值

3.5 数字信号处理初步

数字信号处理技术是一门发展迅速、应用广泛的学科，主要包括信号处理和滤波器设计两部分。MATLAB 提供了专门的信号处理工具箱（Signal Processing Toolbox）和滤波器设计工具箱（Filter Design Toolbox），以满足相关领域用户的需求。

MATLAB 的主平台中提供了一些常用的数字信号处理函数，通过使用这些函数，可以实现数字信号处理的基本功能。

MATLAB 主平台中用于数字信号处理的函数见表 3-4。

表 3-4 数字信号处理的基本函数

函数	功能	函数	功能
abs	复数幅值	fft2	二维快速傅里叶变换
angle	复数相角	filter	数字滤波器
conv	卷积	filter2	二维数字滤波器
con2	二维卷积	ifft	快速傅里叶逆变换
deconv	退卷积	ifft2	二维快速傅里叶逆变换
fft	快速傅里叶变换	fftshift	快速傅里叶结果的重整

3.5.1 快速傅里叶变换

快速傅里叶变换（FFT）是应用最广泛的信号处理方法。$X=\mathrm{fft}(x)$ 和 $x=\mathrm{ifft}(X)$ 实现了对给定的长度为 N 的矢量的变换和逆变换：

$$X(k)=\sum_{j=1}^{N}x(j)\omega_N^{(j-1)(k-1)}$$

$$x(j)=\frac{1}{N}\sum_{k=1}^{N}X(k)\omega_N^{-(j-1)(k-1)}$$

此处 $\omega_N=e^{(-2\pi i)/N}$。

函数 fft 的调用格式为：

• $Y=\mathrm{fft}(X)$ 返回应用快速傅里叶变换方法计算得到的矢量 X 的离散傅里叶变换（DFT）。如果 X 为矩阵，fft 返回矩阵每一列的傅里叶变换；如果 X 为多维数组，fft 运算沿第一个非独立维执行。

• $Y=\mathrm{fft}(X,n)$ 返回 n 点的离散傅里叶变换。如果 X 的长度小于 n，将 X 中补 0 使其与 n 的长度相同；如果 X 的长度大于 n，则 X 的多出部分将被删除；如果 X 为矩阵，同样方法处理矩阵列的长度。

• $Y=\mathrm{fft}(X,[],dim)$ 和 $Y=\mathit{fft}(X,n,dim)$ 沿 dim 维进行 FFT 操作。

需要注意的是，快速傅里叶变换的结果为复数。

【例 3-27】 产生一个正弦衰减曲线，进行快速傅里叶变换，并画出幅值图、相位图、实部图和虚部图。

编辑命令函数，生成正弦衰减曲线，进行快速傅里叶变换，绘制图形显示结果（图 3-9、图 3-10）：

```
tp=0:2048;                              %时域数据点数 N
yt=sin(0.08*pi*tp).*exp(-tp/80);        %生成正弦衰减曲线
t=0:800/2048:800;                       %频域点数 Nf
f=0:1.25:1000;
yf=fft(yt);                             %快速傅里叶变换
ya=abs(yf(1:801));                      %幅值
yp=angle(yf(1:801))*180/pi;             %相位
yr=real(yf(1:801));                     %实部
yi=imag(yf(1:801));                     %虚部
```

```
plot(tp,yt); axis([0,200,-1,1])              %绘制正弦衰减曲线
plot(f,ya), axis([0,200,0,60])               %绘制 FFT 幅值曲线
plot(f,yp), axis([0,200,-200,10])            %绘制 FFT 相位曲线
```

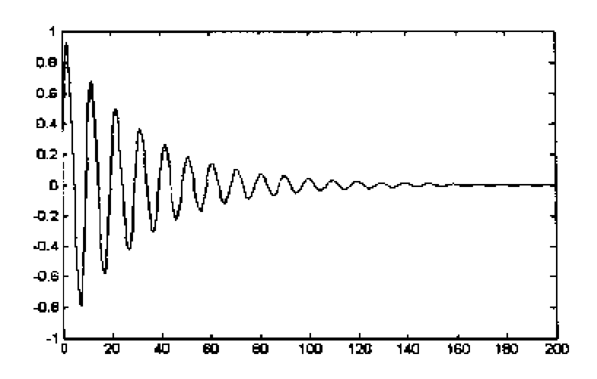


图 3-9　正弦衰减曲线

```
plot(f,yr), axis([0,200,-40,40])             %绘制 FFT 实部曲线
plot(f,yi), axis([0,200,-60,10])             %绘制 FFT 虚部曲线
```

(a) 幅值

(b) 相位

(c) 实部

(d) 虚部

图 3-10　快速傅里叶变换结果

3.5.2 快速傅里叶变换的长度与运算速度

使用 fft 函数时，可以输入第二个参数 n 以指定变换点的数量：

$y = \text{fft}(x, n)$

fft 函数的运算速度取决于变换的长度：

(1) 如果 n 是 2 的整数次幂，则运算速度最快；

(2) 如果 n 是合数，fft 采用质因数分解的算法，速度主要取决于质因数的大小；

(3) 如果 n 是质数，计算速度最慢。

【例 3-28】 比较快速傅里叶变换的长度与运算速度的关系。

创建 70 000×1 阶的随机矢量 x，取快速傅里叶变换的长度分别为质数 65 539、2 的 16 次方 65 536、两个合数 66 000 和 65 535，分别计算使用这些长度计算 fft 所占用的 cpu 时间：

```
>> x=rand(70000,1);
>> isprime(65539)                            %判断 65 539 是否为质数
ans =
    1
>> 2^16                                      %计算 2 的 16 次方
ans =
    65536
>> factor(66000)                             %计算 66 000 的因数分解
ans =
    2     2     2     2     3     5     5     5    11
>> factor(65535)                             %计算 65 535 的因数分解
ans =
    3     5    17   257
>> t=cputime;y=fft(x,65539);e=cputime-t
e =
    0.9900
>> t=cputime;y=fft(x,65536);e=cputime-t
e =
    0.1600
>> t=cputime;y=fft(x,66000);e=cputime-t
e =
    0.1600
>> t=cputime;y=fft(x,65535);e=cputime-t
e =
    0.3800
```

从例 3-28 中可以看出，变换长度的差别对运算速度的影响很大（当然，影响运算速度的因素很多，如计算机本身的性能差异、操作系统、正在运行的其他程序等），因此，正确地选择变换长度是十分重要的，这也正是大多数的数字信号处理系统中将 FFT 的长度取为 2 的 n 次方的原因。

习　题　3

3.1　将$(x-6)(x-3)(x-8)$展开为系数多项式的形式。

3.2　求解多项式 $x^3-7x^2+2x+40$ 的根。

3.3　求解在 $x=8$ 时多项式$(x-1)(x-2)(x-3)(x-4)$的值。

3.4　计算多项式乘法$(x^2+2x+2)(x^2+5x+4)$。

3.5　计算多项式除法$(3x^3+13x^2+6x+8)/(x+4)$。

3.6　对下式进行部分分式展开：

$$\frac{3x^4+2x^3+5x^2+4x+6}{x^5+3x^4+4x^3+2x^2+7x+2}$$

3.7　计算多项式 $4x^4-12x^3-14x^2+5x+9$ 的微分和积分。

3.8　解方程组 $\begin{bmatrix}2 & 9 & 0\\3 & 4 & 11\\2 & 2 & 6\end{bmatrix}x=\begin{bmatrix}13\\6\\6\end{bmatrix}$。

3.9　求欠定方程组 $\begin{bmatrix}2 & 4 & 7 & 4\\9 & 3 & 5 & 6\end{bmatrix}x=\begin{bmatrix}8\\5\end{bmatrix}$ 的最小范数解。

3.10　有一组测量数据如下表所示，数据具有 $y=x^2$ 的变化趋势，用最小二乘法求解 y。

x	1	1.5	2	2.5	3	3.5	4	4.5	5
y	-1.4	2.7	3	5.9	8.4	12.2	16.6	18.8	26.2

3.11　矩阵 $a=\begin{bmatrix}4 & 2 & -6\\7 & 5 & 4\\3 & 4 & 9\end{bmatrix}$，计算 a 的行列式和逆矩阵。

3.12　$y=\sin x$，x 从 0 到 2π，$\Delta x=0.02\pi$，求 y 的最大值、最小值、均值和标准差。

3.13　$x=[1\quad 2\quad 3\quad 4\quad 5]$，$y=[2\quad 4\quad 6\quad 8\quad 10]$，计算 x 的协方差、y 的协方差、x 与 y 的互协方差。

3.14　参照例 3-20 的方法，计算表达式 $z=10\left(x^3-y^5\right)e^{-x^2-y^2}$ 的梯度并绘图。

3.15　有一正弦衰减函数 y=sinx.*exp(-x/10)，其中 x=0:pi/5:4*pi，用三次样条法进行插值。

第 4 章　符号数学基础

在自然科学的各个领域不但需要解决数值分析和计算问题，同时也要解决符号运算的问题。为了解决 MATLAB 在符号数学方面的不足，MathWorks 公司 1993 年从加拿大的滑铁卢大学（University of Waterloo）购入了著名的符号数学软件 Maple 的使用权，并利用 Maple 的函数库，开发了符号数学工具箱。

MATLAB 的符号数学工具箱的主要功能包括：符号表达式的创建、符号矩阵的运算、符号表达式的化简和替换、符号微积分、符号代数方程、符号微分方程、符号函数绘图等。

4.1　符号对象的创建

4.1.1　创建符号变量和表达式

MATLAB 的符号数学工具箱提供了两个基本函数，用来创建符号变量和表达式，分别是 sym 和 syms。

函数 sym 的调用形式为：

- *x*=sym('*x*')　创建一个符号变量 *x*，它可以是字符、字符串、表达式或字符表达式。

函数 syms 用于方便地一次创建多个符号变量，其调用形式为：

- syms *a b c*...

【例 4-1】　使用 sym 函数创建符号变量。

```
>> a=sym('a')
a =
   a
>> b=sym('hello')
b =
   hello
>> c=sym('(1+sqrt(5))/2')
c =
   (1+sqrt(5))/2
>> y=sym('x^3+5*x^2+12*x+20')
y =
   x^3+5*x^2+12*x+20
```

【例 4-2】　使用 syms 函数创建符号变量。

在命令窗口输入：

```
>> syms a b c d
```

由于 syms 函数书写简洁，意义清楚，符合 MATLAB 的习惯和特点，一般提倡使用 syms

创建符号变量。

4.1.2　创建符号矩阵

使用 sym 和 syms 函数可以创建符号矩阵，可以直接输入或从数值矩阵转换。

【例 4-3】　创建一个循环矩阵。

在命令窗口输入：

```
>> syms a b c d
>> n=[a b c d;b c d a;c d a b;d a b c]
n =
    [a, b, c, d]
    [b, c, d, a]
    [c, d, a, b]
    [d, a, b, c]
```

【例 4-4】　将 3 阶 Hilbert 矩阵转换为符号矩阵。

在命令窗口创建 Hilbert 矩阵，然后用 sym 函数转换为符号矩阵。

```
>> h=hilb(3)
h =
    1.0000    0.5000    0.3333
    0.5000    0.3333    0.2500
    0.3333    0.2500    0.2000
>> h1=sym(h)
h1 =
    [  1, 1/2, 1/3]
    [1/2, 1/3, 1/4]
    [1/3, 1/4, 1/5]
```

这里请注意符号矩阵与普通数值矩阵的区别。在命令窗口的显示中，数值矩阵只显示元素的数值，而符号矩阵的每行元素放在一对方括号内；在工作空间窗口显示的变量图标两者也不同，数值矩阵的图标为■，符号矩阵（也称为符号对象）的图标为■，很容易区分。

4.1.3　默认符号变量

按照数学上的习惯，在数学表达式中，字母表中比较靠前的字母表示常量，比较靠后的字母如 x、y、z 表示自变量，例如在表达式 $y = ax^2 + bx + c$ 中，通常认为 x 是自变量，a、b、c 是常量或参数。MATLAB 的符号数学工具箱也采用了类似的规定，在 MATLAB 中，以最接近 x 的顺序排列默认自变量的顺序，可以使用函数 findsym 对默认自变量进行查询。

由于 i 和 j 通常表示虚数单位，在符号运算中不能作为自变量，在 MATLAB 中的符号数学工具箱中，很多运算需要确定表达式中的自变量，不同的自变量会得出不同的结果。

【例 4-5】　求符号函数在不同自变量情况下的结果。

在命令窗口创建符号变量 x 和 n，建立函数 $f = x^n$，然后分别求 f 对 x 和 f 对 n 的导数。

```
>> syms x n
```

```
>> f=x^n
f =
    x^n
>> diff(f)
ans =
    x^n*n/x
>> diff(f,n)
ans =
    x^n*log(x)
```

【例 4-6】 查询符号函数中的默认自变量。

在命令窗口创建符号变量 a、b、n、x 和 t，建立函数 $f = ax^n + bt$，然后求 f 的默认自变量。

```
>> syms a b n t x
>> f=a*x^n+b*t
f =
    a*x^n+b*t
>> findsym(f,1)
ans =
    x
>> findsym(f,2)
ans =
    x,t
>> findsym(f,5)
ans =
    x,t,n,b,a
>> findsym(f)
ans =
    a, b, n, t, x
```

需要注意的是，在例 4-6 中，findsym(*f*, 5) 返回的是 f 表达式中按最接近 x 顺序排列的 5 个默认自变量，findsym(*f*) 返回的是 f 表达式中按字母顺序排列的全部自变量。

4.2 符号表达式的化简和替换

4.2.1 符号表达式的化简

符号数学工具箱提供了符号表达式的因式分解、展开、合并、化简、通分等操作。

1. 因式分解

符号表达式的因式分解函数为 factor，其调用格式为：

- factor(*S*) 因式分解符号表达式 S 的各个元素，如果 S 的所有元素为整数，则计算其最佳因数分解。

【例 4-7】 对表达式 $f = x^9 - 1$ 进行因式分解。

在命令窗口创建符号变量 x，对表达式 $f = x^9 - 1$ 进行因式分解。

```
>> syms x
>> f=factor(x^9-1)
f =
    (x-1)*(x^2+x+1)*(x^6+x^3+1)
>> pretty(f)
                  2             6    3
    (x - 1)(x + x + 1)(x + x + 1)
```

函数 pretty 将符号表达式按照类似书写习惯的方式显示。

【例 4-8】 对大整数 12345678901234567890 进行因式分解。

在命令窗口输入命令，进行因式分解：

```
>> factor(sym('12345678901234567890'))
ans =
    (2)*(3)^2*(5)*(101)*(3803)*(3607)*(27961)*(3541)
```

2. 符号表达式的展开

符号表达式的展开函数为 expand，其调用格式为：

- expand(*S*)　因式展开符号表达式 *S*。

【例 4-9】 展开表达式 $f = (x+1)^5$ 和 $f = \sin(x+y)$。

在命令窗口创建符号变量 x 和 y，展开表达式 $f = (x+1)^5$ 和 $f = \sin(x+y)$：

```
>> syms x y
>> f=(x+1)^5;
>> expand(f)
ans =
    x^5+5*x^4+10*x^3+10*x^2+5*x+1
>> f=sin(x+y);
>> expand(f)
ans =
    sin(x)*cos(y)+cos(x)*sin(y)
```

3. 符号表达式的同类项合并

符号表达式的同类项合并函数为 collect，其调用格式为：

- collect(*S*,*n*)　将符号表达式 *S* 中自变量 *n* 的同次幂项的系数合并。

【例 4-10】 对于表达式 $f = x(x(x-6)+12)t$，分别将自变量 x 和 t 的同类项合并。

在命令窗口创建符号变量 x 和 t，分别按 x 和 t 合并表达式 $f = x(x(x-6)+12)t$ 的同类项：

```
>> syms x t
>> f=x*(x*(x-6)+12)*t;
>> collect(f)
ans =
    t*x^3-6*t*x^2+12*t*x
>> collect(f,t)
ans =
```

```
x*(x*(x-6)+12)*t
```

4. 符号表达式的化简

MATLAB 的符号数学工具箱提供了两个化简函数，分别为 simple 和 simplify。函数 simplify 的调用格式为：

- simplify(*S*) 化简函数，使用 Maple 的化简规则，可用于化简各种表达式。

【例 4-11】 对表达式 $f=\sin^2 x+\cos^2 x$ 进行化简。

在命令窗口创建符号变量 *x*，对表达式 $f=\sin^2 x+\cos^2 x$ 进行化简：

```
>> syms x
>> f=sin(x)^2+cos(x)^2;
>> simplify(f)
ans =
   1
```

函数 simple 通过对表达式尝试多种不同的算法进行化简，以寻求符号表达式 *S* 的最简形式，其调用格式为：

- [*r*, *how*]=simple(*S*) 返回 *S* 的最简化形式，*r* 为返回的简化形式，*how* 为化简过程中使用的主要方法，simple 函数综合使用了下列化简方法：
- simplify 函数对表达式进行化简；
- radsimp 函数对含根式的表达式进行化简；
- combine 函数将表达式中以求和、乘积、幂运算等形式出现的项进行合并；
- collect 合并同类项；
- factor 函数实现因式分解；
- convert 函数完成表达式形式的转换。

表 4-1 列出了使用 simple 函数的一些示例。

表 4-1 使用 simple 函数的一些示例

表达式	简化形式	化简方法
cos(*x*)^2+sin(*x*)^2	1	simplify
2*cos(*x*)^2-sin(*x*)^2	3*cos(*x*)^2−1	simplify
cos(*x*)^2−sin(*x*)^2	cos(2**x*)	combine(trig)
cos(*x*)+(−sin(*x*)^2)^(1/2)	cos(*x*)+*i**sin(*x*)	radsimp
cos(*x*)+i*sin(*x*)	exp(*i***x*)	convert(exp)
(*x*+1)**x**(*x*−1)	*x*^3−*x*	collect(*x*)
x^3+3**x*^2+3**x*+1	(*x*+1)^3	factor
cos(3*acos(*x*))	4**x*^3−3**x*	expand

5. 符号表达式的分式通分

符号表达式的分式通分函数为 numden，其调用格式为：

- [*n*, *d*] = numden(*S*) 将符号表达式 *S* 转换为分子和分母都是整系数的最佳多项式。

【例 4-12】 对表达式 $f=\dfrac{x}{y}+\dfrac{y}{x}$ 进行通分。

在命令窗口创建符号变量 x，对表达式 $f=\dfrac{x}{y}+\dfrac{y}{x}$ 进行通分：

```
>> syms x y
>> f=x/y+y/x;
>> [n,d]=numden(f)
n =
   x^2+y^2
d =
   y*x
```

6. 符号表达式的嵌套形式重写

符号表达式的嵌套形式重写函数为 horner，其调用格式为：

• horner(*S*)　将符号表达式 *S* 转换为嵌套形式。

【例 4-13】　对表达式 $f=x^3+6x^2+11x-6$ 进行嵌套形式重写。

在命令窗口创建符号变量 x，对表达式 $f=x^3+6x^2+11x-6$ 进行嵌套形式重写：

```
>> syms x
>> f=x^3+6*x^2+11*x-6;
>> horner(f)
ans =
    -6+(11+(6+x)*x)*x
```

4.2.2　符号表达式的替换

MATLAB 的符号数学工具箱提供了两个符号表达式的替换函数 subexpr 和 subs，可以通过符号替换使表达式的输出形式简化，以得到一个简单的表达式。

将表达式中重复出现的字符串用变量代替的函数为 subexpr，其调用格式为：

• [*Y*, *SIGMA*] = subexpr (*S*, *SIGMA*)　此函数用变量 *SIGMA*（字符或字符串）的值代替符号表达式 *S* 中重复出现的字符串，*Y* 返回替换后的结果。

【例 4-14】　求解并化简三次方程 $x^3+ax+1=0$ 的符号解。

在命令窗口创建符号表达式 t = solve('x^3+a*x+1 = 0')，并通过符号替换进行化简：

```
>> t = solve('x^3+a*x+1 = 0')
t =
    [1/6*(-108+12*(12*a^3+81)^(1/2))^(1/3)-2*a/(-
         108+12*(12*a^3+81)^(1/2))^(1/3)]
    [-1/12*(-108+12*(12*a^3+81)^(1/2))^(1/3)+a/
         (-108+12*(12*a^3+81)^(1/2))^(1/3)+1/2*i*3^(1/2)*(1/
         6*(-108+12*(12*a^3+81)^(1/2))^(1/3)+2*a/
         (-108+12*(12*a^3+81)^(1/2))^(1/3))]
    [-1/12*(-108+12*(12*a^3+81)^(1/2))^(1/3)+a/
         (-108+12*(12*a^3+81)^(1/2))^(1/3)-1/2*i*3^(1/2)*(1/
         6*(-108+12*(12*a^3+81)^(1/2))^(1/3)+2*a/
         (-108+12*(12*a^3+81)^(1/2))^(1/3))]
>> [r,s] = subexpr(t,'s')
```

```
r =
    [                                   1/6*s^(1/3)-2*a/s^(1/3)]
    [-1/12*s^(1/3)+a/s^(1/3)+1/2*i*3^(1/2)*(1/6*s^(1/3)+2*a/s^(1/3))]
    [-1/12*s^(1/3)+a/s^(1/3)-1/2*i*3^(1/2)*(1/6*s^(1/3)+2*a/s^(1/3))]
s =
    -108+12*(12*a^3+81)^(1/2)
```

函数 subs 用指定符号替换符号表达式中的某一特定符号，其调用格式为：

• R = subs (S, *old*, *new*)　用新的符号变量 *new* 替换原来符号表达式 S 中的变量 *old*。需要注意的是，当变量 *new* 是数值形式时，显示的结果虽然是数值，但它事实上仍然是符号变量。

【例 4-15】 分别用新变量替换表达式 $a+b$ 和 $\cos a+\sin b$ 中的变量。

在命令窗口创建符号变量 a 和 b，输入变量替换命令：

```
>> syms a b
>> subs(a+b,a,4)
ans =
    4+b
>> subs(cos(a)+sin(b),{a,b},{sym('alpha'),2})
ans =
    cos(alpha)+sin(2)
```

4.3 符号微积分

4.3.1 符号极限

极限是微积分的基础。在 MATLAB 中，极限的求解由 limit 函数实现，其调用格式为：

• limit(F, x, a)　计算符号表达式 F 在 $x \to a$ 条件下的极限。

• limit(F, a)　计算符号表达式 F 中默认自变量趋向于 a 条件下的极限。

• limit(F)　计算符号表达式 F 在默认自变量趋向于 0 时的极限。

• limit(F, x, a, '*right*') 和 limit(F, x ,a, '*lift*')　计算符号表达式 F 在 $x \to a$ 条件下的右极限和左极限。

【例 4-16】 分别计算表达式 $\lim\limits_{x\to 0}\dfrac{\sin x}{x}$、$\lim\limits_{x\to 0_+}\dfrac{1}{x}$、$\lim\limits_{x\to 0_-}\dfrac{1}{x}$ 及 $\lim\limits_{x\to\infty_-}\left(1+\dfrac{a}{x}\right)^x$ 和 $\lim\limits_{x\to\infty_-} e^{-x}$ 的极限。

在命令窗口创建符号变量 a 和 x，分别计算上面各表达式的极限：

```
>> syms x a;
>> limit(sin(x)/x)
ans =
    1
>> limit(1/x,x,0,'right')
ans =
    inf
>> limit(1/x,x,0,'left')
```

```
ans =
    -inf
>> v = [(1 + a/x)^x, exp(-x)];
>> limit(v,x,inf,'left')
ans =
    [ exp(a),      0]
```

4.3.2　符号微分

在符号数学工具箱中，表达式的微分由函数 diff 实现，其调用格式为：

- diff(*S*)　求符号表达式 *S* 对于默认自变量的微分。
- diff(*S*, *v*)　求符号表达式 *S* 对于自变量 *v* 的微分。
- diff(*S*, *n*)　求符号表达式 *S* 对于默认自变量的 *n* 次微分。

【例 4-17】　分别求表达式 $f = x^x$ 的导数和 3 次导数。

在命令窗口创建符号变量 *x*，计算 $f = x^x$ 的导数和 3 次导数：

```
>> syms x;
>> f=x^x;
>> diff(f)
ans =
    x^x*(log(x)+1)
>> diff(f,3)
ans =
    x^x*(log(x)+1)^3+3*x^x*(log(x)+1)/x-x^x/x^2
```

4.3.3　符号积分

在符号数学工具箱中，表达式的积分由函数 int 实现，其调用格式为：

- int(*S*)　求符号表达式 *S* 对于默认自变量的不定积分。
- int(*S*,*v*)　求符号表达式 *S* 对于自变量 *v* 的不定积分。
- int(*S*,*a*,*b*)　求符号表达式 *S* 对于默认自变量从 *a* 到 *b* 的定积分。

【例 4-18】　分别计算表达式 $\int \frac{-2x}{(1+x^2)^2}\mathrm{d}x$、$\int \frac{x}{(1+z)^2}\mathrm{d}x$、$\int \frac{x}{(1+z)^2}\mathrm{d}z$ 和 $\int_0^1 x\log(1+x)\mathrm{d}x$。

在命令窗口创建符号变量 *x* 和 *z*，分别计算上面各表达式的积分：

```
>> syms x z;
>> f=-2*x/(1+x^2)^2;
>> int(f)
ans =
    1/(1+x^2)
>> f=x/(1+z^2);
>> int(f)
ans =
    1/2*x^2/(1+z^2)
```

```
>> int(f,z)
ans =
   x*atan(z)
>> f=x*log(1+x);
>> int(f,0,1)
ans =
   1/4
```

4.3.4 符号求和

在符号数学工具箱中，表达式的求和由函数 symsum 实现，其调用格式为：

- symsum(*S*) 计算符号表达式 *S* 对于默认自变量的不定和。
- symsum(*S*,*v*) 计算符号表达式 *S* 对于自变量 *v* 的不定和。
- symsum(*S*,*a*,*b*) 计算符号表达式 *S* 对于默认自变量从 *a* 到 *b* 的有限和。

【例 4-19】 分别计算表达式 $\sum k$、$\sum_{0}^{10} k^2$ 和 $\sum_{k=0}^{\infty}\frac{x^k}{k!}$。

在命令窗口创建符号变量 *k* 和 *x*，分别计算上面各表达式：

```
>> syms k x
>> symsum(k)
ans =
   1/2*k^2-1/2*k
>> symsum(k^2,0,10)
ans =
   385
>> symsum(x^k/sym('k!'), k, 0,inf)
ans =
   exp(x)
```

4.3.5 Taylor 级数展开

在符号数学工具箱中，表达式的 Taylor 级数展开由函数 taylor 实现，其调用格式为：

- taylor (*f*) 计算符号表达式 *f* 在默认自变量等于 0 处的 5 阶 Taylor 级数展开式。
- taylor (*f*, *n*, *v*) 计算符号表达式 *f* 在自变量 $v=0$ 处的 $n-1$ 阶 Taylor 级数展开式。
- taylor (*f*, *n*, *v*, *a*) 计算符号表达式 *f* 在自变量 $v=a$ 处的 $n-1$ 阶 Taylor 级数展开式。

【例 4-20】 分别计算表达式 $f=\dfrac{1}{5+\cos x}$ 的 5 阶 Taylor 级数展开式和 $f=\mathrm{e}^{x\sin x}$ 的 5 阶及 12 阶 Taylor 级数展开式。

在命令窗口创建符号变量 *x*，分别计算上面各表达式：

```
>> syms x
>> f=1/(5+cos(x));
>> r=taylor(f)
r =
```

```
    1/6+1/72*x^2
>> f=exp(x*sin(x));
>> r=taylor(f)
r =
    1+x^2+1/3*x^4
>> r=taylor(f,12)
r =
    1+x^2+1/3*x^4+1/120*x^6-11/560*x^8-1079/362880*x^10
```

MATLAB 的符号数学工具箱还提供了图形化的 Taylor 级数工具，其功能和使用在本书 4.7 中介绍。

4.4　符号方程的求解

4.4.1　符号代数方程组的求解

在符号数学工具箱中，求解表达式的代数方程由函数 solve 实现，其调用格式为：

- g = solve(*eq*)　求解符号表达式 *eq* = 0 的代数方程，自变量为默认自变量。
- g = solve(*eq*, *var*)　求解符号表达式 *eq* = 0 的代数方程，自变量为 *var*。
- g = solve(*eq*1, *eq*2, …, *eqn*, *var*1, *var*2, …, *varn*)　求解符号表达式 *eq*1，*eq*2，…，*eqn* 组成的代数方程组，自变量分别为 *var*1，*var*2，…，*varn*。

【例 4-21】　分别求解代数方程 $ax^2+bx+c=0$ 和 $\cos 2x+\sin x=1$。

在命令窗口创建符号变量 *a*、*b*、*c* 和 *x*，分别求解各方程：

```
>> syms a b c x
>> s=a*x^2+b*x+c;
>> solve(s)
ans =
    [1/2/a*(-b+(b^2-4*a*c)^(1/2))]
    [1/2/a*(-b-(b^2-4*a*c)^(1/2))]
>> solve('cos(2*x)+sin(x)=1')
ans =
    [       0]
    [      pi]
    [1/6*pi]
    [5/6*pi]
```

需要注意的是，由于三角函数是周期函数，往往不能解出某一区间的全部解。

【例 4-22】　求解代数方程组 $\begin{cases} x^2-y^2+z=10 \\ x+y-5z=0 \\ 2x-4y+z=0 \end{cases}$。

在命令窗口创建符号变量 *x*、*y*、*z*，求解方程组：

```
>> syms x y z
```

```
>> f=x^2-y^2+z-10;
>> g=x+y-5*z;
>> h=2*x-4*y+z;
>> [x,y,z]=solve(f,g,h)
x =
    [-19/80+19/240*2409^(1/2)]
    [-19/80-19/240*2409^(1/2)]
y =
    [-11/80+11/240*2409^(1/2)]
    [-11/80-11/240*2409^(1/2)]
z =
    [-3/40+1/40*2409^(1/2)]
    [-3/40-1/40*2409^(1/2)]
>> S=solve(f,g,h);
>> [S.x,S.y,S.z]
ans =
    [-19/80+19/240*2409^(1/2),-11/80+11/240*2409^(1/2),
        -3/40+1/40*2409^(1/2)]
    [-19/80-19/240*2409^(1/2),-11/80-11/240*2409^(1/2),
        -3/40-1/40*2409^(1/2)]
```

在MATLAB 6中，方程组的解缺省情况存放在结构变量中，即本例中的*S*，也可以定义数值数组，如本例中的$[x, y, z]$。

4.4.2 符号微分方程求解

在符号数学工具箱中，求表达式常微分方程的符号解由函数dsolve实现，其调用格式为：

- *r* = dsolve('*eq*1, *eq*2, ...', '*cond*1, *cond*2, ...', '*v*')　求由*eq*1，*eq*2，...指定的常微分方程的符号解，参数*cond*1，*cond*2，...为指定常微分方程的边界条件或初始条件，自变量*v*如果不指定，将为默认自变量。

在方程中，用大写字母*D*表示一次微分，*D*2和*D*3分别表示二次及三次微分，*D*后面的字符为因变量。

【例4-23】　求微分方程$\frac{dy}{dt} = ay$的通解和当$y(0) = b$时的特解。

在命令窗口分别输入表达式，求解方程：

```
>> dsolve('Dy = a*y')
ans =
    C1*exp(a*t)
>> dsolve('Dy = a*y', 'y(0) = b')
ans =
    b*exp(a*t)
```

【例4-24】　求微分方程$\frac{d^2y}{d^2t} = -a^2y$当$y(0) = 1$及$\frac{dy}{dt}\left(\frac{\pi}{a}\right) = 0$时的特解。

在命令窗口输入表达式，求解方程：

```
>> dsolve('D2y = -a^2*y', 'y(0) = 1', 'Dy(pi/a) = 0')
ans =
    cos(a*t)
```

4.5　符号数学的简易绘图函数

MATLAB 提供了一系列简易绘图函数，这些函数的功能和作用与 MATLAB 的普通绘图函数基本相同，但简易绘图函数使用极为简单，一般只要在简易绘图函数的参数中指定所绘制的函数名即可。当然，与普通绘图函数相比，简易绘图函数的功能相对比较简单，但对于显示函数的图形来说，已经足够了。

4.5.1　二维绘图函数

二维绘图函数 plot 是 MATLAB 最基本和最常用的绘图函数，其对应的简易绘图函数为 ezplot。ezplot 的调用格式为：

- ezplot(*f*)　绘制表达式 *f*(*x*)的二维图形，*x* 轴坐标的近似范围为[−2π, 2π]。
- ezplot(*f*, [*xmin*, *xmax*])　绘制表达式 *f*(*x*)的二维图形，*x* 轴坐标的范围为[*xmin*, *xmax*]。

【例 4-25】　绘制函数表达式 x^2-y^4 的二维图形。

在命令窗口输入表达式，绘制图形（图 4-1）：

```
>> syms x y
>> ezplot(x^2-y^4)
```

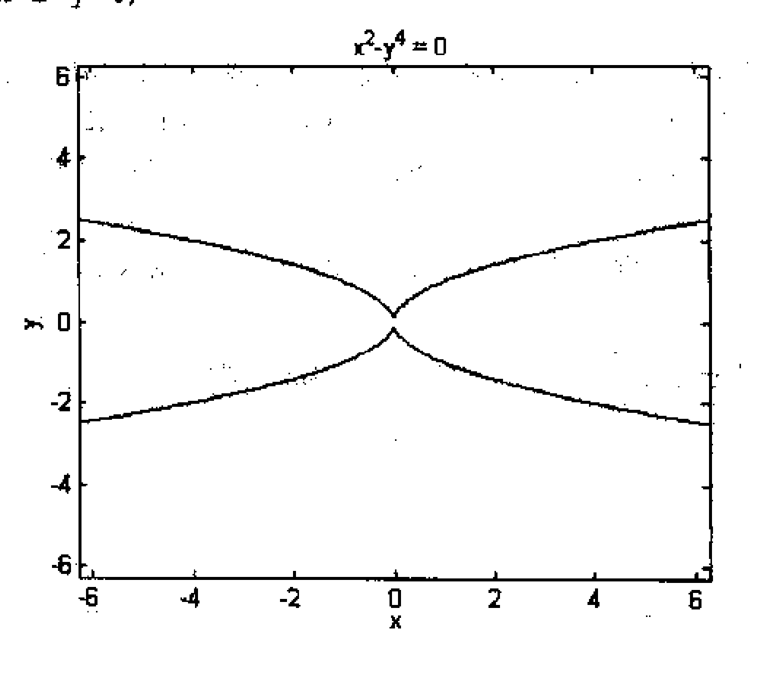


图 4-1　二维绘图函数

【例 4-26】　绘制误差函数 $f(x)=\frac{2}{\sqrt{\pi}}\int_0^x \mathrm{e}^{-t^2}\mathrm{d}t$ 的二维图形。

在命令窗口输入表达式，绘制图形，grid 命令为绘制网格（图 4-2）：

```
>> syms x
>> ezplot(erf(x))
>> grid
```

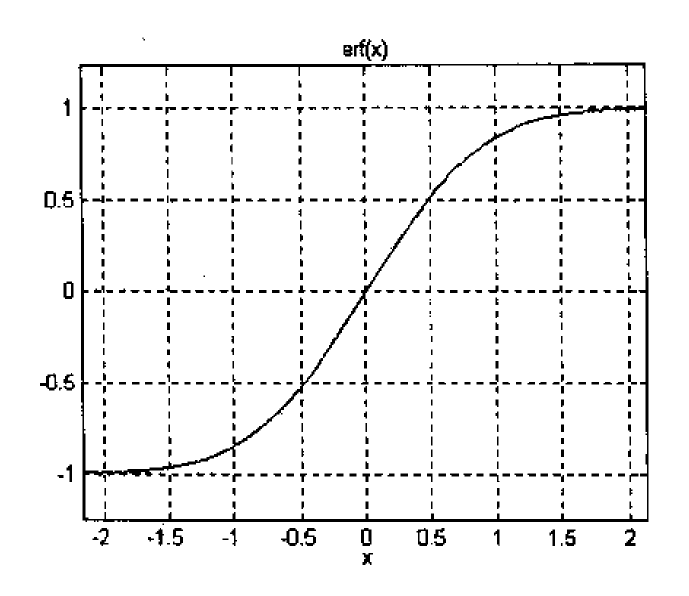


图 4-2 二维绘图函数（带网格）

ezpolar 函数为极坐标下的二维绘图函数，调用格式与 ezplot 相同。

【例 4-27】 在极坐标下绘制函数表达式 $1+\cos t$ 的二维图形。

在命令窗口输入表达式，绘制图形（图 4-3）：

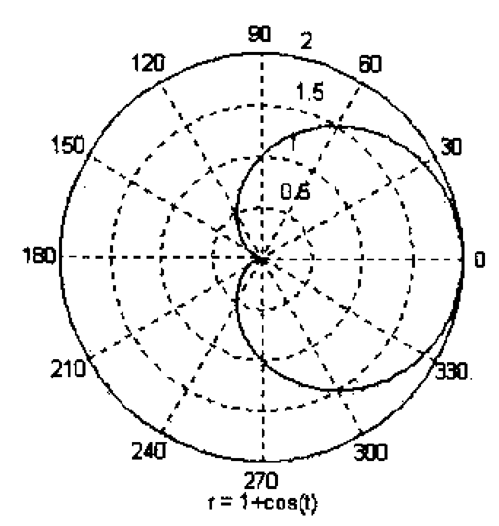


图 4-3 二维极坐标绘图函数

```
>> syms t
>> ezpolar(1+cos(t))
```

4.5.2　三维绘图函数

三维曲线图的简易绘图函数为 ezplot3，其调用格式为：

• ezplot3(*x*, *y*, *z*)　绘制由表达式 $x = x(t)$、$y = y(t)$ 和 $z = z(t)$ 定义的三维曲线，自变量 *t* 的变化范围为[−2π, 2π]。

• ezplot3(*x*, *y*, *z*, [*tmin*, *tmax*])　绘制由表达式 $x = x(t)$、$y = y(t)$ 和 $z = z(t)$ 定义的三维曲线，自变量 *t* 的变化范围为[*tmin*, *tmax*]。

• ezplot3(..., 'animate')　如果在函数中增加 animate 参数，则绘制三维动态轨迹图。

【例 4-28】　根据表达式 $x = \sin t$、$y = \cos t$ 和 $z = t$，绘制三维曲线。

在命令窗口输入表达式，绘制三维曲线（图 4-4）：

```
>> syms t;
>> ezplot3(sin(t), cos(t), t,[0,6*pi])
```

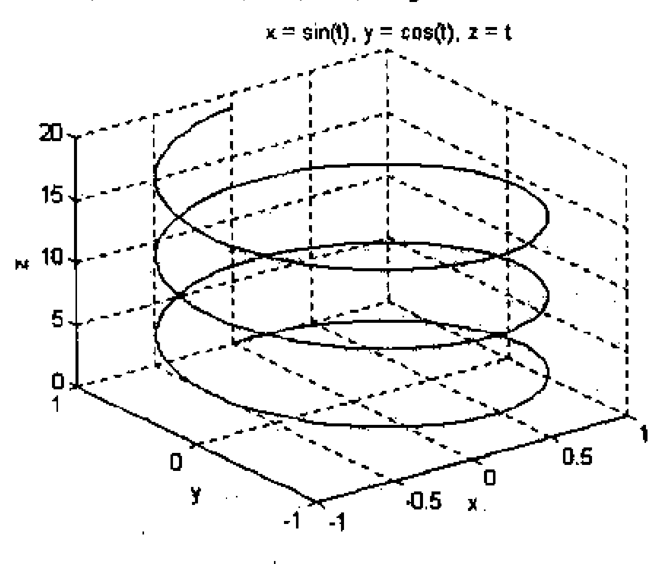


图 4-4　三维曲线绘图函数

4.5.3　等高线绘图函数

等高线的简易绘图函数为 ezcontour，其调用格式为：

• ezcontour(*f*)　绘制由表达式 *f*(*x*, *y*) 定义的等高线，自变量 *x* 和 *y* 的变化范围均为[−2π, 2π]。

• ezcontour(*f*, *domain*)　绘制由表达式 *f*(*x*, *y*) 定义的等高线，自变量 *x* 和 *y* 的变化范围由 *domain* 确定，*domain* 可以是 4×1 阶的矢量 [*xmin*, *xmax*, *ymin*, *ymax*]，也可以是 2×1 阶的矢量[min, max]，当 *domain* 为 2×1 阶的矢量时，min < *x* < max，min < *y* < max。

• ezcontour(..., *n*)　绘制等高线时按 *n*×*n* 的网格密度绘图，*n* 的缺省值为 60。

【例 4-29】　根据表达式 $f = 3(1-x)^2 \mathrm{e}^{-x^2-(y+1)^2} - 10\left(\frac{x}{5} - x^3 - y^5\right)\mathrm{e}^{-x^2-y^2} - \frac{1}{3}\mathrm{e}^{-(x+1)^2-y^2}$，绘制 *f* 的等高线。

在命令窗口输入表达式，绘制等高线（图 4-5）：

```
>> syms x y
>> f =3*(1-x)^2*exp(-(x^2)-(y+1)^2)-10*(x/5-x^3-y^5)...
     *exp(-x^2-y^2)-1/3*exp(-(x+1)^2- y^2);
>> ezcontour(f,[-3,3],49)
```

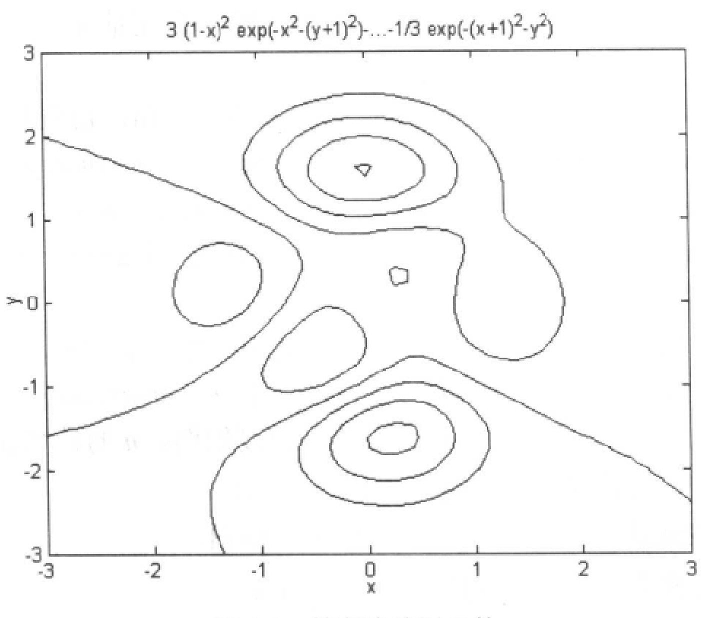


图 4-5 等高线绘图函数

填充等高线的简易绘图函数为 ezcontourf，其调用格式与 ezcontour 相同。

【例 4-30】 根据表达式 $f=3(1-x)^2\mathrm{e}^{-x^2-(y+1)^2}-10\left(\dfrac{x}{5}-x^3-y^5\right)\mathrm{e}^{-x^2-y^2}-\dfrac{1}{3}\mathrm{e}^{-(x+1)^2-y^2}$，绘制 f 的填充等高线。

在命令窗口输入表达式，绘制填充等高线（图 4-6）：

```
>> syms x y
>> f =3*(1-x)^2*exp(-(x^2)-(y+1)^2)-10*(x/5-x^3-y^5)...
     *exp(-x^2-y^2)-1/3*exp(-(x+1)^2- y^2);
>> ezcontourf(f,[-3,3],49)
```

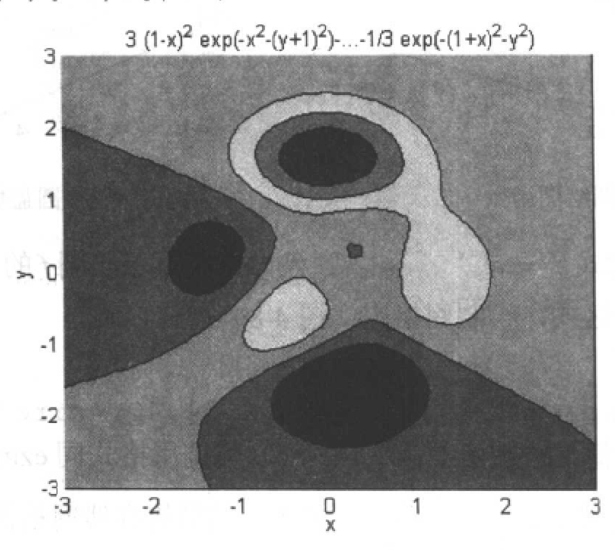


图 4-6 填充等高线绘图函数

4.5.4 网格图绘图函数

网格图的简易绘图函数为 ezmesh，其调用格式为：

- ezmesh (*f*)　绘制由表达式 $f(x, y)$ 定义的网格图，自变量 x 和 y 的变化范围均为$[-2\pi, 2\pi]$。
- ezmesh(*f*, *domain*)　绘制由表达式 $f(x, y)$定义的网格图，自变量 x 和 y 的变化范围由 *domain* 确定，*domain* 可以是 4×1 阶的矢量 [*xmin*, *xmax*, *ymin*, *ymax*]，也可以是 2×1 阶的矢量[min, max]，当 *domain* 为 2×1 阶的矢量时，min < x < max,，min < y < max。
- ezmesh (*x*, *y*, *z*)　绘制由表达式 $x = x(s, t)$、$y = y(s, t)$ 和 $z = z(s, t)$ 定义的参数表面网格图，自变量 s 和 t 的变化范围均为$[-2\pi, 2\pi]$。
- ezmesh (*x*, *y*, *z*, [*smin*, *smax*, *tmin*, *tmax*])　绘制由表达式 $x = x(s,t)$、$y = y(s,t)$ 和 $z = z(s, t)$ 定义的参数表面网格图，自变量 s 和 t 的变化范围为[*smin*, *smax*, *tmin*, *tmax*]。
- ezmesh (..., *n*)　绘制网格图时按 $n\times n$ 的网格密度绘图，n 的缺省值为 60。
- ezmesh(..., 'circ')　以圆盘为自变量域绘制网格图。

【例 4-31】　根据表达式 $f = xe^{-x^2-y^2}$，绘制 f 的网格图。

在命令窗口输入表达式，绘制网格图（图 4-7）：

```
>> syms x y
>> ezmesh(x*exp(-x^2-y^2),[-2.5,2.5],40)
>> colormap([0 0 1])
```

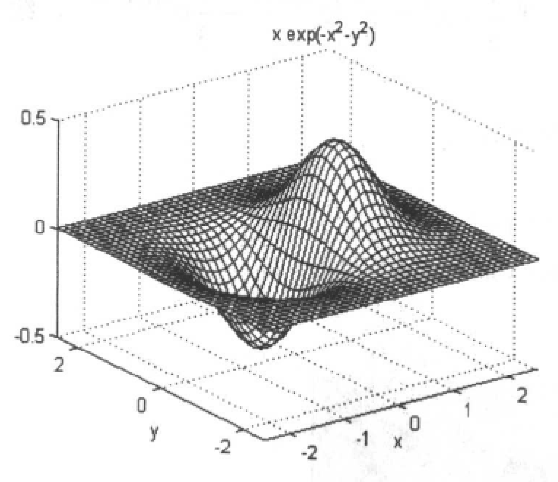


图 4-7　网格图绘图函数

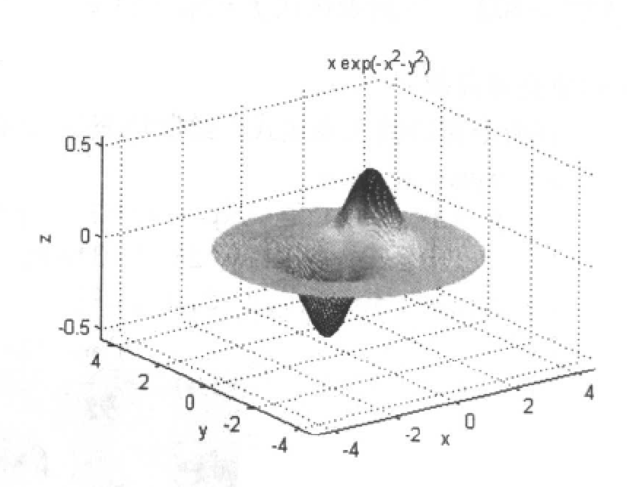


图 4-8　圆盘域网格图绘图函数

【例 4-32】　根据表达式 $f = xe^{-x^2-y^2}$，以圆盘为自变量域绘制 f 的网格图。

在命令窗口输入表达式，绘制网格图（图 4-8）：

```
>> syms x y
>> ezmesh(x*exp(-x^2-y^2),[-2.5,2.5],40,'circ')
```

带等高线网格图的简易绘图函数为 ezmeshc，其调用格式同 ezmesh。

【例 4-33】　根据表达式 $f = \dfrac{y}{1+x^2+y^2}$，绘制 f 的带等高线网格图。

在命令窗口输入表达式，绘制带等高线网格图（图 4-9）：

```
>> syms x y
>> ezmeshc(y/(1 + x^2 + y^2),[-5,5,-2*pi,2*pi])
```

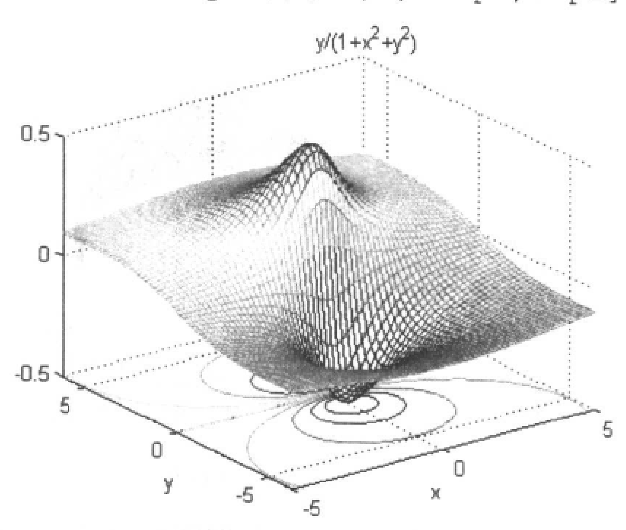


图 4-9 带等高线网格图绘图函数

4.5.5 表面图绘图函数

表面图的简易绘图函数为 ezsurf，其调用格式为：

• ezsurf(*f*) 绘制由表达式 $f(x, y)$ 定义的表面图，自变量 x 和 y 的变化范围均为 $[-2\pi, 2\pi]$。

• ezsurf (*f*, *domain*) 绘制由表达式 $f(x, y)$ 定义的表面图，自变量 x 和 y 的变化范围由 *domain* 确定，*domain* 可以是 4×1 阶的矢量 [*xmin*, *xmax*, *ymin*, *ymax*]，也可以是 2×1 阶的矢量[min, max]，当 *domain* 为 2×1 阶的矢量时，$\min < x < \max$，$\min < y < \max$。

• ezsurf (*x*, *y*, *z*) 绘制由表达式 $x = x(s, t)$、$y = y(s, t)$ 和 $z = z(s, t)$ 定义的参数表面图，自变量 s 和 t 的变化范围均为 $[-2\pi, 2\pi]$。

• ezsurf (*x*, *y*, *z*, [*smin*, *smax*, *tmin*, *tmax*]) 绘制由表达式 $x = x(s, t)$、$y = y(s, t)$ 和 $z = z(s, t)$ 定义的参数表面图，自变量 s 和 t 的变化范围为[*smin*, *smax*, *tmin*, *tmax*]。

• ezsurf (..., *n*) 绘制表面图时按 $n \times n$ 的网格密度绘图，n 的缺省值为 60。

• ezsurf (..., 'circ') 以圆盘为自变量域绘制表面图。

【例 4-34】 根据表达式 $x = \cos s \cos t$ 、 $y = \cos s \sin t$ 和 $z = \sin s$ ，绘制表面图。

在命令窗口输入表达式，绘制表面图（图 4-10）：

```
>> syms t s
>> x=cos(s)*cos(t);
>> y=cos(s)*sin(t);
>> z=sin(s);
>> ezsurf(x,y,z,[0,pi/2,0,3*pi/2])
>> view(17,40)
>> shading interp
```

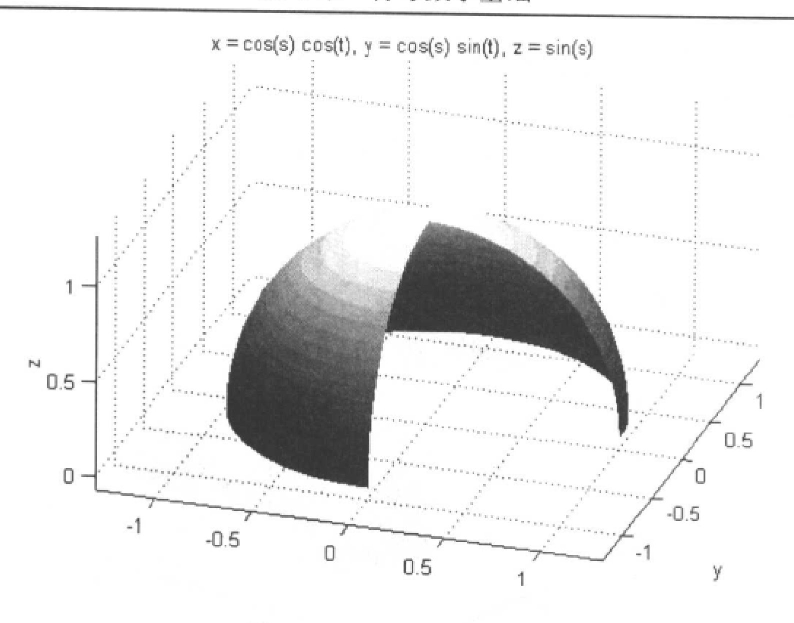


图 4-10　表面图绘图函数

带等高线表面图的简易绘图函数为 ezsurfc，其调用格式同 ezsurf。

【例 4-35】　根据表达式 $f=\dfrac{y}{1+x^2+y^2}$，绘制 f 带等高线的表面图。

在命令窗口输入表达式，绘制带等高线表面图（图 4-11）：

```
>> syms x y
>> ezsurfc(y/(1 + x^2 + y^2),[-5,5,-2*pi,2*pi],35)
>> view(-65,26)
```

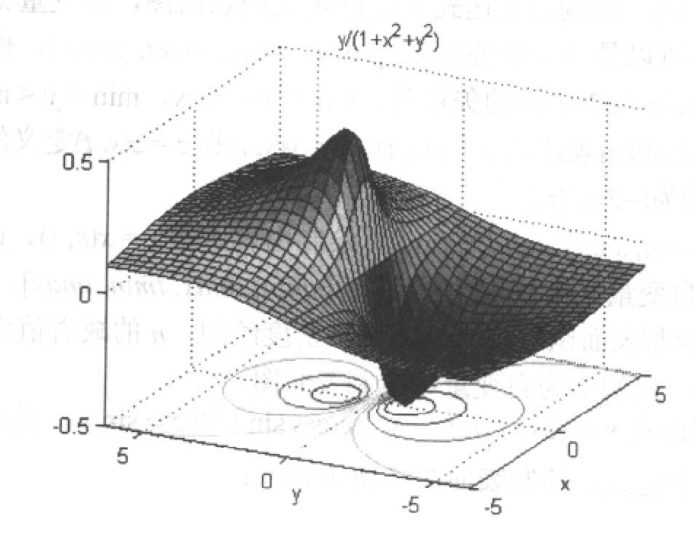


图 4-11　表面图绘图函数

4.6　图形化符号函数计算器

MATLAB 的符号数学工具箱中提供了一个图形化符号函数计算器函数 funtool，它具有功能简单实用、操作方便的优点，同时由于具有图形界面，为用户提供了直观的工具。

在 MATLAB 的命令窗口输入 funtool，即可启动图形化符号函数计算器，其界面如图 4-12 所示。

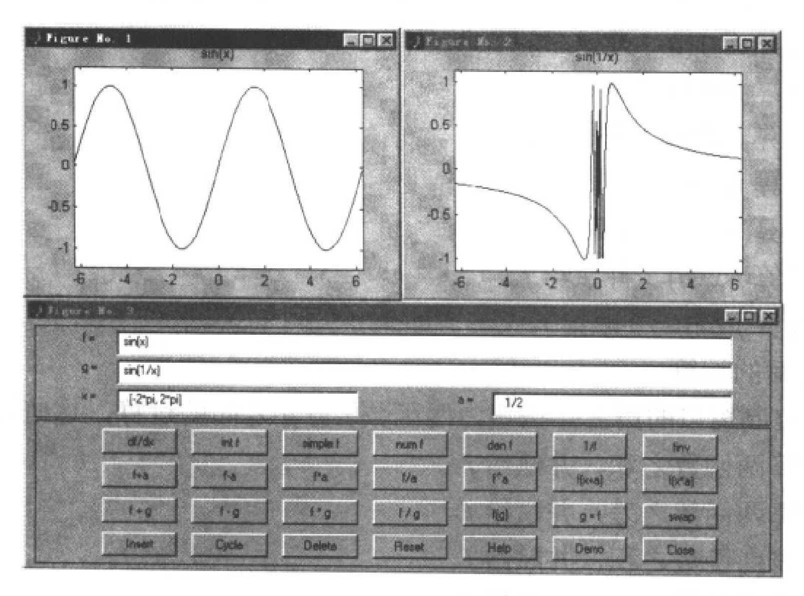


图 4-12 图形化符号函数计算器

图形化符号函数计算器由三个窗口组成，分别为两个图形窗口和一个函数功能的控制窗口，图形窗口 1 是函数 f 的显示窗口，图形窗口 2 是函数 g 的显示窗口。下面详细介绍图形化符号函数计算器的使用。

4.6.1 输入框

在控制窗口的上方有 4 个输入框，用户可以在输入框中输入函数和自变量等参数，4 个输入框分别为：

- f= 为图形窗口 1 的控制函数，其缺省值为 x；
- g= 为图形窗口 2 的控制函数，其缺省值为 1；
- x= 为两窗口函数的自变量取值范围，缺省为$[-2\pi, 2\pi]$；
- a= 为常数值，缺省为 1/2。

4.6.2 计算器的功能

1. 函数的自运算

函数功能的控制窗口的第一排按钮为函数的自运算，这些运算包括：

- df/dx 计算函数 f 对 x 的导函数，并赋给 f。
- int f 计算函数 f 的积分函数，并赋给 f。
- simple f 计算函数 f 的最简表达式，并赋给 f。
- num f 取表达式 f 的分子，并赋给 f。
- den f 取表达式 f 的分母，并赋给 f。
- 1/f 求 f 的倒数函数，并赋给 f。
- finv 求 f 的反函数，并赋给 f。

2. 函数与常数的运算

函数功能的控制窗口的第二排按钮为函数与常数之间的运算，这些运算包括：

- f+a　　计算 $f(x)+a$，并赋给 f。
- f−a　　计算 $f(x)-a$，并赋给 f。
- f*a　　计算 $f(x)*a$，并赋给 f。
- f/a　　计算 $f(x)/a$，并赋给 f。
- f^a　　计算 $(f(x))^a$，并赋给 f。
- f(x+a)　计算 $f(x+a)$，并赋给 f。
- f(x*a)　计算 $f(ax)$，并赋给 f。

3. 两函数之间的运算

函数功能的控制窗口的第三排按钮为两函数 f 与 g 之间的运算，这些运算包括：

- f+g　　计算两函数之和，并赋给 f。
- f−g　　计算两函数之差，并赋给 f。
- f*g　　计算两函数之积，并赋给 f。
- f/g　　计算两函数之比，并赋给 f。
- f(g)　　求复合函数 $f(g(x))$。
- g=f　　将 f 的函数值赋给 g。
- swap　　交换 f 与 g 的函数表达式。

4. 函数计算器的系统操作

函数功能的控制窗口的第四排按钮为函数计算器的系统操作，这些操作包括：

- insert　　将当前窗口 1 中的函数加到计算器的典型函数表中。
- cycle　　在窗口 1 依次演示计算器典型函数表中的函数。
- delete　　从计算器的典型函数表中删除当前窗口 1 中的函数。
- reset　　符号函数计算器的功能重置。
- help　　符号函数计算器的在线帮助。
- demo　　符号函数计算器功能演示（演示中相应功能按钮变白）。
- close　　关闭符号函数计算器。

4.7 Taylor 级数计算器

Taylor 级数计算器是 MATLAB 符号数学工具箱中提供的另一个图形化符号函数可视化工具，用于显示 Taylor 级数展开的精度。

在 MATLAB 的命令窗口输入 taylortool，即可启动图形化的 Taylor 级数计算器，其界面如图 4-13 所示。

使用 Taylor 级数计算器可以非常直观地观察 Taylor 级数展开与原函数和接近程度，以选择合适的 Taylor 级数。

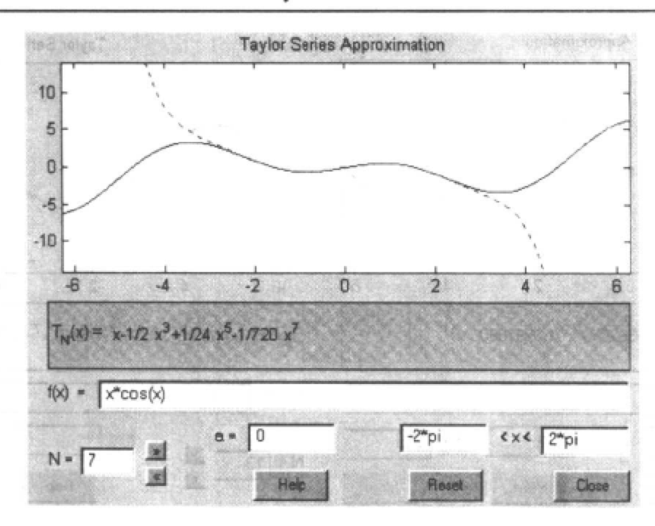


图 4-13 Taylor 级数计算器

【例 4-36】 使用 Taylor 级数计算器。

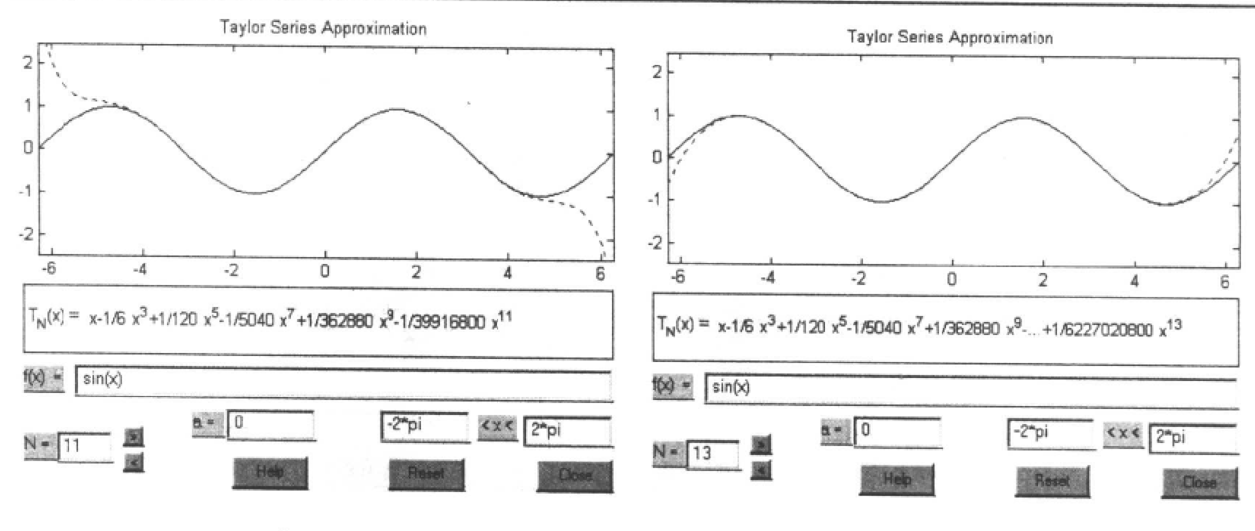
对 y=sin$x$ 进行 Taylor 级数展开，观察各阶展开的逼近程度（图 4-14），表 4-2 列出了各阶展开系数。

(a) 3 阶 Taylor 级数展开

(b) 5 阶 Taylor 级数展开

(c) 7 阶 Taylor 级数展开

(d) 9 阶 Taylor 级数展开



(e)　11 阶 Taylor 级数展开　　　　(f)　13 阶 Taylor 级数展开

图 4-14　各阶 Taylor 级数展开

表 4-2　各阶 Taylor 级数展开系数

阶数	展开系数
3	$x-\frac{1}{6}x^3$
5	$x-\frac{1}{6}x^3+\frac{1}{120}x^5$
7	$x-\frac{1}{6}x^3+\frac{1}{120}x^5-\frac{1}{5040}x^7$
9	$x-\frac{1}{6}x^3+\frac{1}{120}x^5-\frac{1}{5040}x^7+\frac{1}{362880}x^9$
11	$x-\frac{1}{6}x^3+\frac{1}{120}x^5-\frac{1}{5040}x^7+\frac{1}{362880}x^9-\frac{1}{39916800}x^{11}$
13	$x-\frac{1}{6}x^3+\frac{1}{120}x^5-\frac{1}{5040}x^7+\frac{1}{362880}x^9-\frac{1}{39916800}x^{11}+\frac{1}{6227020800}x^{13}$

习　题　4

4.1　创建符号变量有几种方法？

4.2　下面三种表示方法有什么不同的含义？

(1) f = 3*x^2+5*x+2

(2) f = '3*x^2+5*x+2'

(3) x = sym('x')

　f = 3*x^2+5*x+2

4.3　用符号函数法求解方程 $at^2+b*t+c=0$。

4.4　用符号计算验证三角等式：

$\sin\varphi_1\cos\varphi_2-\cos\varphi_1\sin\varphi_2=\sin(\varphi_1-\varphi_2)$

4.5　求矩阵 $A=\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ 的行列式值、逆和特征根。

4.6　因式分解 $x^4-5x^3+5x^2+5x-6$。

4.7 $f=\begin{bmatrix} a & x^2 & \dfrac{1}{x} \\ e^{ax} & \log x & \sin x \end{bmatrix}$，用符号微分求 df/dx。

4.8 求代数方程组 $\begin{cases} ax^2+by+c=0 \\ x+y=0 \end{cases}$ 关于 x、y 的解。

4.9 用符号函数绘图法绘制函数 $x=\sin(3t)\cos(t)$，$y=\sin3t\sin t$ 的图形，t 的变化范围为[0, 2π]。

4.10 绘制极坐标下 sin(3*t)*cos*t* 的图形。

4.11 使用 Taylor 级数计算器求 *x*cos*x* 的 Taylor 级数展开，并比较展开的接近程度。

第 5 章　基本图形处理功能

MATLAB 中提供了强大的图形处理功能，应用 MATLAB 可以实现各种二维图形和三维图形的绘制和控制与表现。本章介绍 MATLAB 的基本图形处理功能。

5.1　二 维 图 形

5.1.1　基本绘图函数

1. 二维曲线绘图

二维曲线绘图函数 plot 是最基本的绘图函数，其调用格式为：

• plot(*Y*)　绘制矢量 *Y* 对于其元素序数的二维曲线图，如果 *Y* 为复数，则绘制虚部对于实部的二维曲线图。

• plot(*X*1, *Y*1,...)　绘制所有由 *X*1，*Y*1，...所确定的曲线，具体使用方法将在 5.1.2 中介绍。

• plot(*X*1, *Y*1, *LineSpec*,...)　按照 *LineSpec* 所定义的线型、标记点和颜色绘图，*LineSpec* 的具体使用方法将在 5.1.3 中介绍。

• plot(..., '*PropertyName*', *PropertyValue*, ...)　为所有由 plot 函数创建的图形对象设置属性值。

• *h* = plot(...)　返回一个图形对象句柄。

【例 5-1】　绘制单矢量曲线图。

在命令窗口输入矢量并绘图（图 5-1）：

```
>> y=[0 0.6 2.3 5 8.3 11.7 15 17.7 19.4 20];
>> plot(y)
```

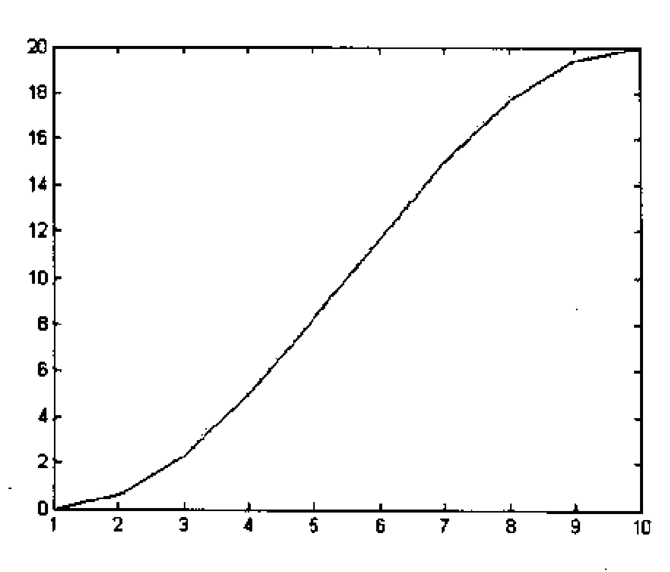


图 5-1　单矢量输入的曲线图

单矢量绘图是一种最简单的绘图方法，只需对 y 矢量赋值即可进行绘图。MATLAB 会自动给出图中坐标轴的范围和刻度，由于 y 矢量共有 10 个元素，则 x 坐标自动定义为 [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]。

【例 5-2】 绘制双矢量曲线图。

在命令窗口输入两个矢量并绘图（图 5-2）：

```
>> x=0:0.05:4*pi;          %生成 0 至 4π，间隔为 0.05 的自变量
>> y=sin(x);
>> plot(x,y)
```

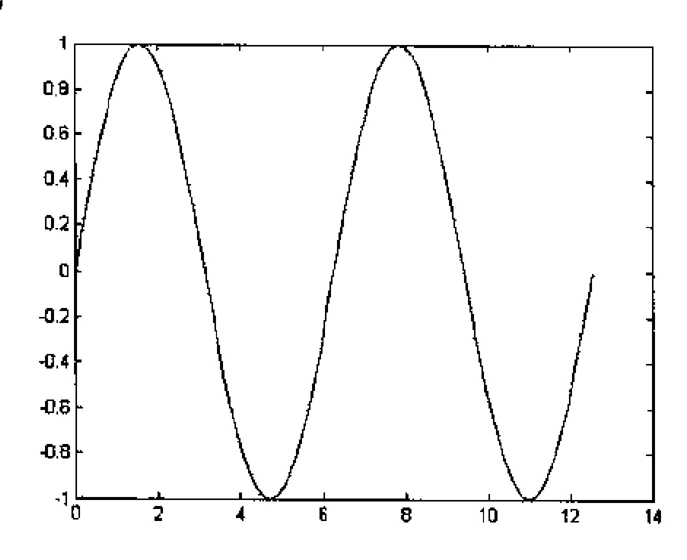


图 5-2 双矢量输入的曲线图

注意，绘制双矢量曲线图时，矢量 x 与 y 必须具有同样的长度。

2. 对数坐标绘图

对数坐标绘图分为 x 轴对数（semilogx）、y 轴对数（semilogy）及双对数（loglog）3 个函数，其调用格式与二维曲线绘图函数 plot 相似。

【例 5-3】 使用例 5-1 的数据绘制线性坐标图和 3 种对数坐标图。

在命令窗口输入矢量，同时绘制线性坐标图和 3 种对数坐标图（图 5-3）：

```
>> y=[0 0.6 2.3 5 8.3 11.7 15 17.7 19.4 20];
>> subplot(2,2,1)                                %创建第 1 个子图
>> plot(y)
>> title('plot(y)')
>> subplot(2,2,2)                                %创建第 2 个子图
>> semilogx(y)
>> title('semilogx(y)')
>> subplot(2,2,3)                                %创建第 3 个子图
>> semilogy(y)
>> title('semilogy(y)')
>> subplot(2,2,4)                                %创建第 4 个子图
>> loglog(y)
>> title('loglog(y)')
```

函数 subplot 用于绘制多个子图，命令 title 用于在图形上书写标题，它们的具体用法见

5.4。

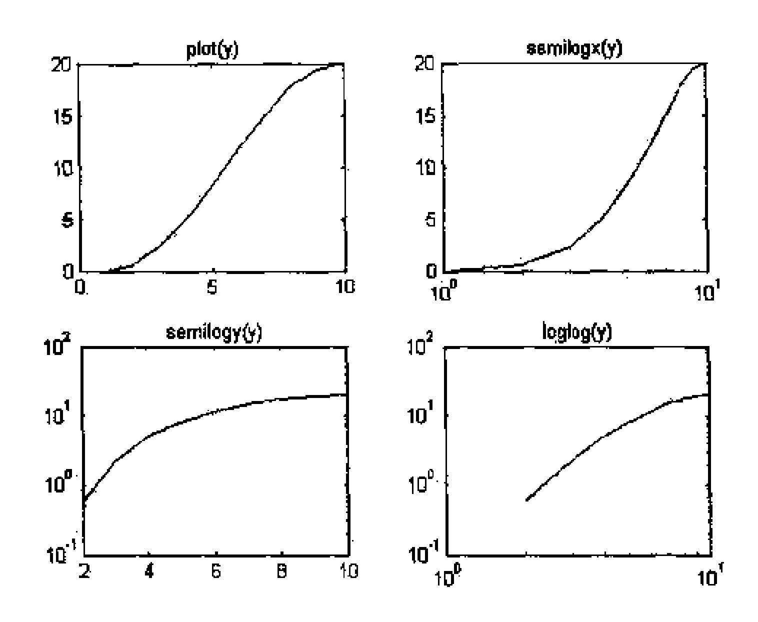


图 5-3　线性及 3 种对数坐标下的曲线图

3. 极坐标绘图

极坐标绘图的函数为 polar，其调用格式为：

• polar(*theta*, *rho*)　绘制由 *theta*、*rho* 所确定的曲线，这里绘图参数 *theta* 和 *rho* 分别为角度和半径。

• polar(*theta*, *rho*, *LineSpec*)　按照 *LineSpec* 所定义的线型、标记点和颜色绘制极坐标图，*LineSpec* 的具体使用方法将在 5.1.3 中介绍。

【例 5-4】　绘制函数 $y = \sin 2t\cos 2t$ 的极坐标图形。

在命令窗口创建角度矢量 t，以函数 $y = \sin 2t\cos 2t$ 为半径绘图（图 5-4）：

```
>> t=0:.01:2*pi;
>> polar(t,sin(2*t).*cos(2*t))
```

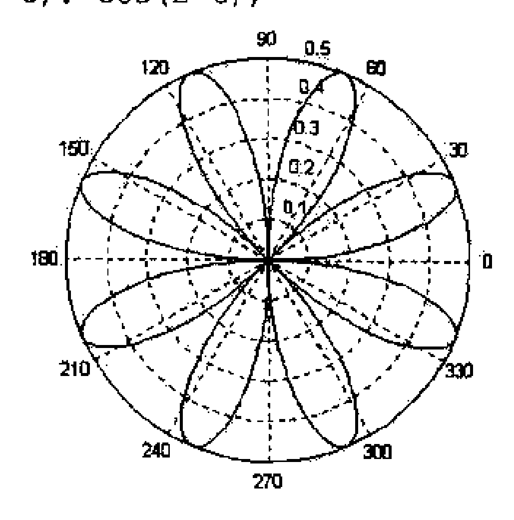


图 5-4　极坐标曲线图

4. 复数绘图

当使用 plot 函数绘制复数图形时，通常虚部是忽略的。然而有一种特殊情况，即当 plot

的参数只有一个复变量 z 时，则绘制的图形为实部对于虚部的关系图（复平面上的一组点）。这时 plot(z)等价于 plot(real(z), imag(z))，其中 z 为一个复矢量。

【例 5-5】　绘制 20 边的多边形，顶角用小圆圈表示。

在命令窗口创建矢量 t，用 exp 函数生成圆周并绘图（图 5-5）：

```
>> t=0:pi/10:2*pi;
>> plot(exp(i*t),'-o')
>> axis('square')
```

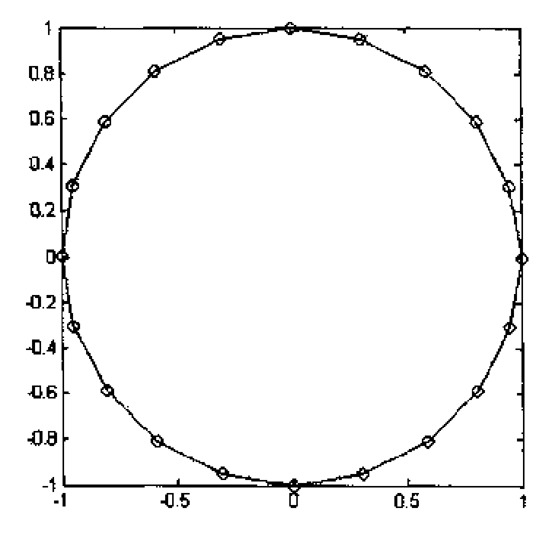
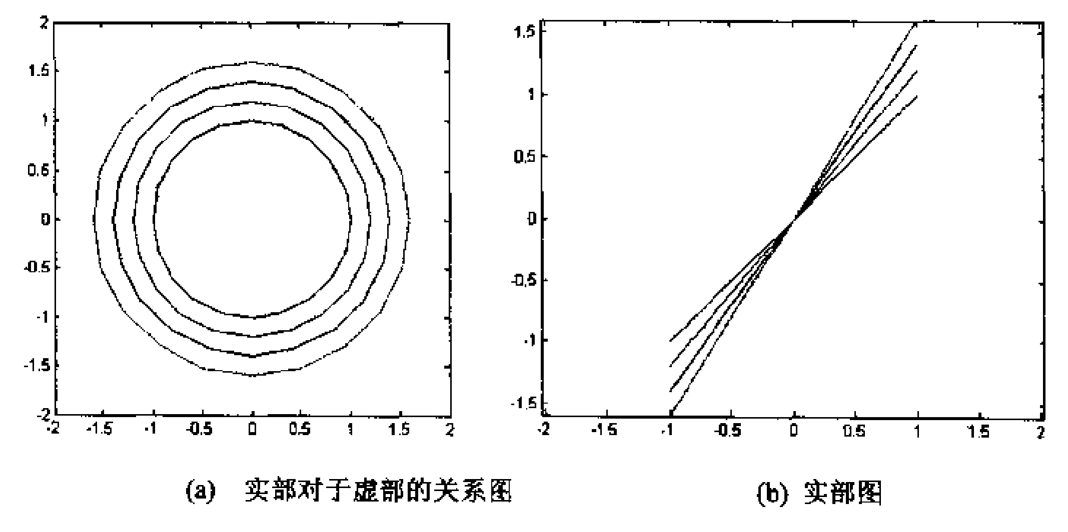


图 5-5　复矢量绘图

【例 5-6】　复矩阵绘图比较。

在命令窗口创建矢量 t 和 x，然后创建矩阵 y，绘制复平面图和实部图（图 5-6）：

```
>> t=0:pi/10:2*pi;
>> x=exp(i*t)';
>> y=[x 1.2*x 1.4*x 1.6*x];
>> plot(y); axis('square')          %在复平面绘制实部对于虚部的关系图
>> plot(x,y); axis('equal')         %在实平面绘制实部图，虚部被忽略
Warning: Imaginary parts of complex X and/or Y arguments ignored.
```



(a)　实部对于虚部的关系图　　　(b) 实部图

图 5-6　复矩阵的绘图

5.1.2 多重曲线绘图

使用 plot 函数在单幅图形上绘制多重曲线有两种方法。一种方法是用一组变量 x 和 y 绘图，其中 x 或 y 是矩阵，或两个都是矩阵。另一种方法是用多组变量 x_1、y_1、x_2、y_2、…、x_n、y_n 绘图。在绘制多重曲线时，MATLAB 会按照一定的规律自动变化每条曲线的颜色。

1. 一组变量绘图

用一组变量 x 和 y 绘图分为以下几种情况：

(1) 如果 x 是矢量，y 是矩阵，plot(x, y)用不同颜色的图线绘制 y 矩阵中各行或列对应于 x 的曲线；如果 y 矩阵中行的长度与 x 矢量的长度相同，MATLAB 就以行数据作为一组绘图数据；如果 y 矩阵中列的长度与 x 矢量的长度相同，则以列数据作为一组绘图数据。如果 y 是一个行列均与 x 的长度相等的方阵，则以每列作为一组绘图数据。

【例 5-7】　当 x 为矢量，y 为矩阵时绘制多重曲线图。

在命令窗口创建矢量 x，然后创建函数矩阵 y 并绘图（图 5-7）：

```
>> x=0:pi/50:2*pi;
>> y(1,:)=sin(x);
>> y(2,:)=0.6*sin(x);
>> y(3,:)=0.3*sin(x);
>> plot(x,y)
```

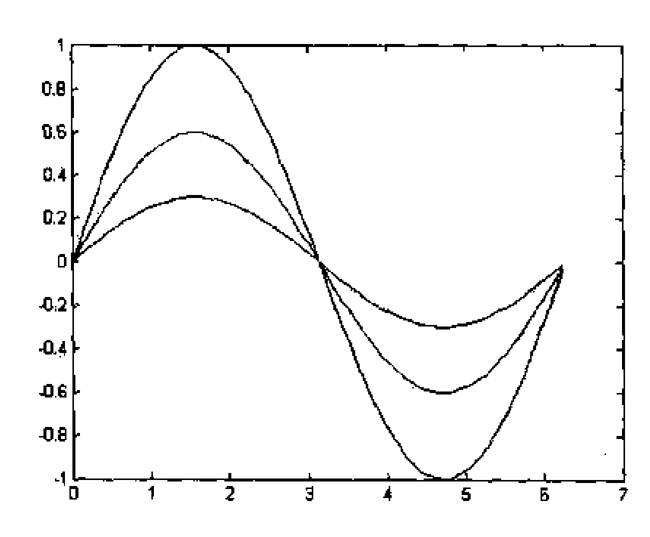


图 5-7　x 为矢量 y 为矩阵时的多重曲线图

(2) 如果 x 是矩阵，y 是矢量，则按类似 (1) 的规则，只是将 x 中的每一行或列对应于 y 进行绘图。

【例 5-8】　当 x 为矩阵，y 为矢量时绘制多重曲线图。

在命令窗口创建矩阵 x 和矢量 y 并绘图（图 5-8）：

```
>> x(1,:)=0:pi/50:2*pi;
>> x(2,:)=pi/4:pi/50:2*pi+pi/4;
>> x(3,:)=pi/2:pi/50:2*pi+pi/2;
>> y=sin(x(1,:));
>> plot(x,y)
```

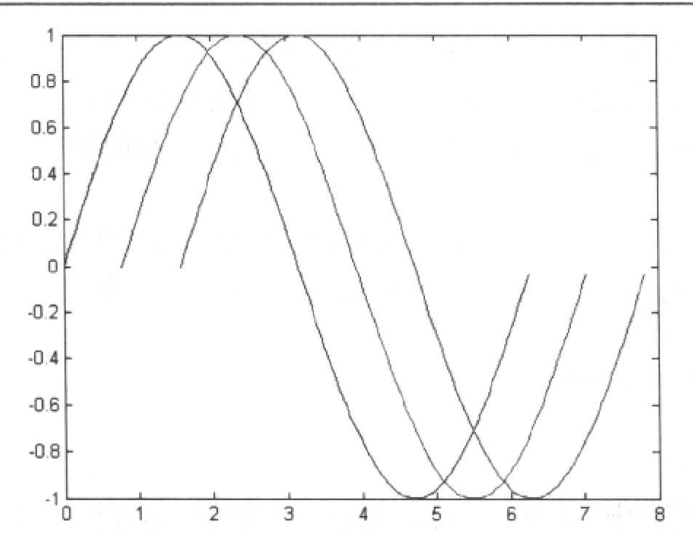


图 5-8　*x* 为矩阵 *y* 为矢量时的多重曲线图

（3）如果 *x*、*y* 是同样大小的矩阵，则 plot(*x*, *y*)绘出 *y* 中各列相应于 *x* 中各列的图形。

【例 5-9】　当 *x* 和 *y* 为同样大小的矩阵时，绘制多重曲线图。

在命令窗口创建矩阵 *x* 和 *y* 并绘图（图 5-9）：

```
>>x(1,:)=0:pi/50:2*pi;
>>x(2,:)=pi/4:pi/50:2*pi+pi/4;
>>x(3,:)=pi/2:pi/50:2*pi+pi/2;
>>y(1,:)=sin(x(1,:));
>>y(2,:)=0.6*sin(x(1,:));
>>y(3,:)=0.3*sin(x(1,:));
>>plot(x,y)
>>x=x';
>>y=y';
>>figure                                   %创建新的图形窗口
>>plot(x,y)
```

(a) 数据矩阵为 3×101　　　　　　　　(b) 数据矩阵为 101×3

图 5-9　*x* 与 *y* 都是矩阵的曲线图

这里要注意的是，图 5-9（a）中的 *x* 与 *y* 的尺寸都是 3×101，所以按列画出 101 条曲线，

每条曲线是 3 个点。图 5-9（b）中的 x 与 y 的尺寸都是 101×3（为了对比，数据内容没有改变，只是做了转置），所以按列画出 3 条曲线，每条曲线是 101 个点。

(4) 如果 y 是矩阵，则 plot(y)绘制出 y 中各列相对于行号的图形，对于 n 行矩阵，x 轴的坐标为[1:n]。

对于 x 与 y 中有一个为矢量的情况，矢量的长度必须与矩阵中行或列的长度相等；对于 x 与 y 都是矩阵的情况，两矩阵的行与列的长度必须都相等。如果不满足上述条件，绘图时系统会输出错误提示：

```
??? Error using ==> plot
Vectors must be the same lengths.
```

2. 多组变量绘图

对于一系列相应的矩阵 yi 和 xi，可以使用多组变量绘图方法，即

plot($x1, y1, x2, y2, \cdots, xn, yn$)

这种用法把 $y1$ 对于 $x1$、$y2$ 对于 $x2$、…等的图线绘在一幅图形上，而且可以分别采用不同的图线形式。这种多组变量绘图的优点是允许将不同大小的矩阵或矢量的图形绘制在一幅图形上。

【例 5-10】　使用多组变量，绘制多重曲线图。

在命令窗口创建矩阵 x 和 y 并绘图（图 5-10）：

```
>> x1=0:pi/50:2*pi;                     %生成 100 个数据点
>> x2=0:pi/30:2*pi;                     %生成 60 个数据点
>> x3=0:pi/15:2*pi;                     %生成 30 个数据点
>> y1=sin(x1);
>> y2=0.6*sin(x2);
>> y3=0.3*sin(x3);
>> plot(x1,y1,x2,y2,x3,y3)
```

将例 5-7 与例 5-10 进行比较可以看到，两个例子的差别在于数据是否在一个数组中以及数组长度的不同，两种方法绘出的图形基本上是一样的。两种方法各有优点，对于曲线较多或使用直接按数组计算得到的数据时适合用单数组绘图，如果希望控制每条曲线的线型、颜色、标记点等特性以及绘制不同矢量（矩阵）长度的数据时适合用多组变量绘图。

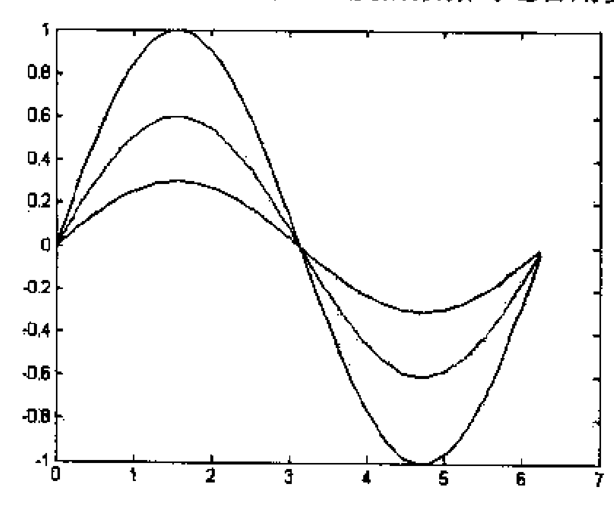


图 5-10　多组变量绘图

3. 双 Y 轴绘图

MATLAB 提供了绘制双坐标轴曲线的函数 plotyy，其调用格式为：

• plotyy(X1, Y1, X2, Y2)　绘制由 X1、Y1 和 X2、Y2 确定的两组曲线，其中 X1、Y1 的坐标轴在图形窗口的左侧，X2、Y2 的坐标轴在图形窗口的右侧。

• plotyy(X1, Y1, X2, Y2, 'function')　使用 function 指定的绘图函数绘制由 X1、Y1 和 X2、Y2 确定的两组曲线，function 可以是 plot、semilogx、semilogy、loglog、stem 或所有接受以下语法的 MATLAB 函数：

h = function(x, y)

• plotyy(X1, Y1, X2, Y2, 'function1', 'function2')　对于图形窗口左侧的坐标轴使用 function1(X1, Y1)绘图，对于图形窗口右侧的坐标轴使用 function2(X2, Y2)绘图。

• [AX, H1, H2] = plotyy(...)　返回 AX 中创建的两个坐标轴的句柄以及 H1 和 H2 中每个图形绘图对象的句柄，AX(1)为左侧轴，AX(2)为右侧轴。

【例 5-11】　在一个图形窗口中绘制双 Y 轴曲线。

在命令窗口创建矩阵 x 和 y，使用双 Y 轴方式绘图：

```
>> x=0:0.3:12;
>> y=exp(-0.3*x).*sin(x)+0.5;
>> plotyy(x,y,x,y,'plot','stem')
```

绘图结果见图 5-11，两条图线自动用不同的颜色区分，两个坐标的颜色与图线颜色相对应，左边的 y 坐标对应的是 plot 形式的图线，右边的 y 坐标对应的是 stem 形式的图线（关于 stem 函数的用法见 5.3.3）。

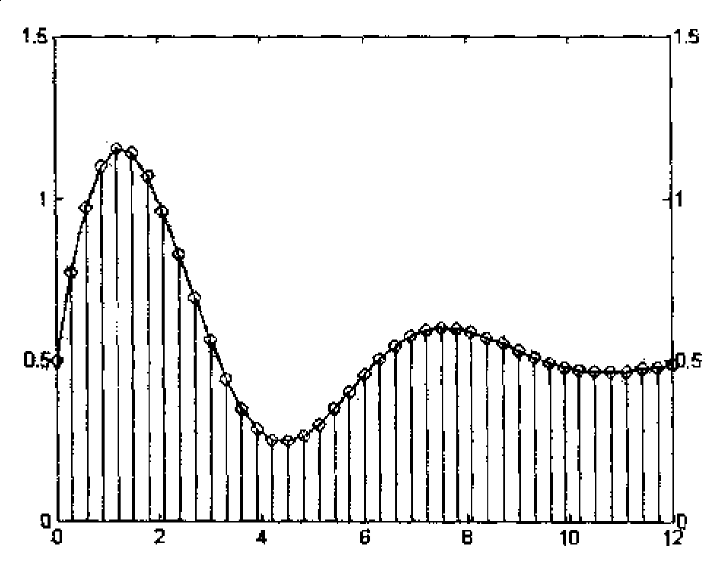


图 5-11　双 Y 坐标轴绘图（Y 坐标相同）

【例 5-12】　使用不同的 Y 坐标，在一个图形窗口中绘制两组数据曲线。

在命令窗口创建矩阵 t、$z1$ 和 $z2$，使用两种绘图方式绘制双 Y 轴图。

```
>> t=0:900;  A=1000;  a=0.005;  b=0.005;
>> z1=A*exp(-a*t);
>> z2=sin(b*t);
>> plotyy(t,z1,t,z2,'semilogy','plot')
```

从程序中可以看到 z1 为指数曲线，z2 为正弦曲线，分别使用半对数坐标和线性坐标绘图，绘图结果见图 5-12。

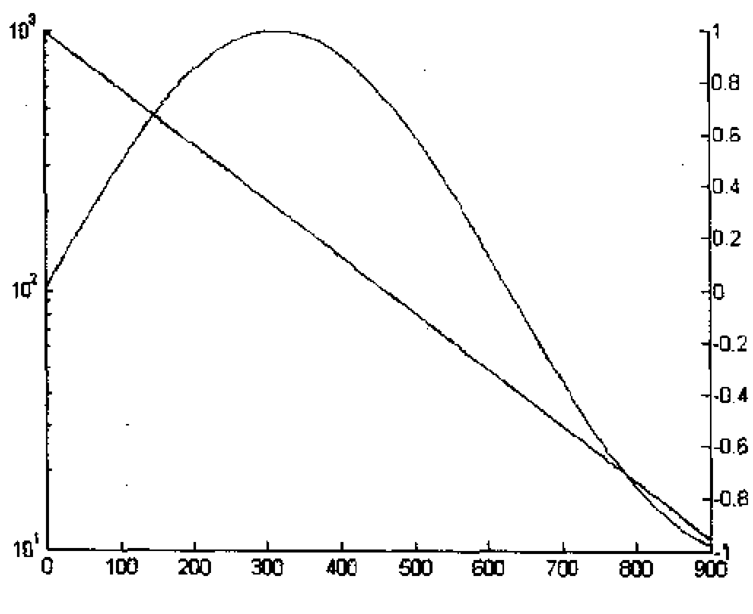


图 5-12　双 Y 坐标轴绘图（Y 坐标不同）

5.1.3　图线形式和颜色

1. 图线的形式

MATLAB 的曲线绘图有两种形式，分别为线绘图方式和点绘图方式。线绘图方式是将数据点用某种形式的线连接起来，在线绘图方式中可以选择线型，缺省线型为实线；点绘图方式是只把数据点用某种类型的标记点绘制出来而不连线，在点绘图方式中可以选择标记点的类型，根据需要也可以将两种绘图方式结合使用。设置图线的一般格式为：

• plot(x, y, '--')　用虚线把数据点连接起来，绘制虚折线图形。

• plot(x1, y1, ':', x2, y2, '*')　用“:”线型绘制第一条由 x1 和 y1 确定的曲线，用“*”点方式绘制第 2 条由 x2 和 y2 确定的曲线。

MATLAB 中提供了 4 种线型和 13 种标记点类型，分别见表 5-1 和表 5-2。

在绘图时指定线型和标记点的方法非常简单，只要将表示线型和标记点的符号用字符的形式作为 plot 函数中的一个输入参数即可。

表 5-1　线型的选择

线型	符号	示例
实线	-	——————
虚线	--	— — — — —
冒号线	:	···············
点划线	-.	— · — · — · —

表 5-2　标记点类型的选择

点类型	符号	示例
点	.	· · · · ·
加号	+	+ + + + +
星号	*	* * * * *
圆圈	o	○ ○ ○ ○ ○
叉号	x	× × × × ×
正方形	s(或 square)	□ □ □ □ □
菱形	d(或 diamond)	◇ ◇ ◇ ◇ ◇
上三角	^	△ △ △ △ △
下三角	v	▽ ▽ ▽ ▽ ▽
左三角	<	◁ ◁ ◁ ◁ ◁
右三角	>	▷ ▷ ▷ ▷ ▷
五角星	p(或 pentagram)	☆ ☆ ☆ ☆ ☆
六角星	h(或 hexagram)	✡ ✡ ✡ ✡ ✡

【例 5-13】　使用不同的线型绘图。

在命令窗口创建矢量 t、$y1$、$y2$ 和 $y3$，使用不同的线型绘图（图 5-13）：

```
>>t=0:pi/100:2*pi;
>>y=sin(t);
>>y2=sin(t-0.25);
>>y3=sin(t-0.5);
>>plot(t,y,'-',t,y2,'-.',t,y3,':')
```

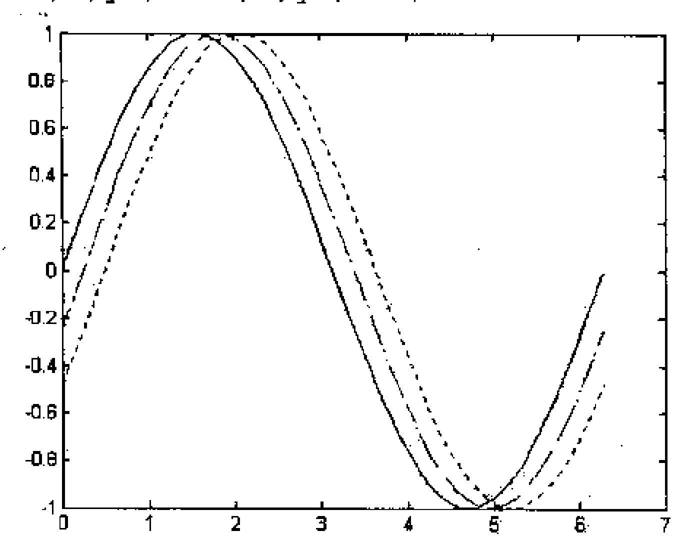


图 5-13　使用不同的绘图线型

说明：在 plot 函数中，第 1 组数据后面的'-'表示用实线绘制第 1 条线，第 2 组数据后面的'-.'表示用点划线绘制第 2 条线，第 3 组数据后面的':'表示用短虚线绘制第 3 条线。

【例 5-14】　使用不同的标记点绘图。

在命令窗口创建矢量 t 和 y，使用不同的标记点绘图（图 5-14）：

```
>> t=0:pi/20:2*pi;
>> y=sin(t);
>> y2=sin(t-0.25);
>> y3=sin(t-0.5);
>> plot(t,y,'o',t,y2,'*',t,y3,'+')
```

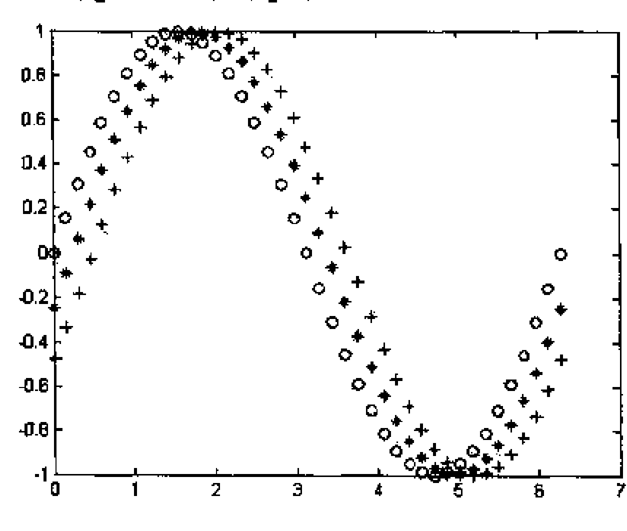


图 5-14　使用不同的标记点绘图

2. 图线的颜色

图形中的线和标记点可以设置为各种不同的颜色，MATLAB 基本绘图函数中可选择的颜色如表 5-3 所示。

表 5-3　可以选择的颜色

颜色	符号	颜色	符号
红	r	粉红	m
绿	g	青	c
蓝	b	白	w
黄	y	黑	k

指定线和标记点的颜色的方法与指定线型和标记点的方法相同。基本调用格式为：

- plot(x, y, 'r')　绘制的曲线为红色。
- plot(x, y, 'g+')　绘制的曲线为绿色，且使用“+”作为标记点。

在指定线型、颜色和标记点三种属性时应注意：(1) 3 种属性的符号必须放在同一个字符串中；(2) 可以只指定其中的一个属性，也可以同时指定两个或 3 个属性；(3) 属性的先后顺序无关；(4) 指定的属性中同种属性不能有两个以上。

3. 图线的其他属性

除了设定线型、标记点和颜色等基本属性外，MATLAB 基本绘图函数还可以进一步设置包括线的宽度、标记点的边缘颜色、填充颜色及标记点的大小等其他绘图属性。

在 MATLAB 的基本绘图函数的参数中，除了数据点和指定线型、颜色及标记点的符号串以外，还可以包括线宽（LineWidth）、标记点边缘颜色（MarkerEdgeColor）、标记点填

充颜色（MarkerFaceColor）和标记点大小（MarkerSize）等属性。

【例 5-15】 设置绘图线的线型、颜色、宽度、标记点的颜色及大小。

在命令窗口创建矩阵 t 和 y，设置绘图线的线型、颜色、宽度、标记点的颜色及大小（图 5-15）：

```
>> t=0:pi/20:pi;
>> y=sin(4*t).*sin(t)/2;
>> plot(t,y,'-bs','LineWidth',2,...          %设置线的宽度为 2
        'MarkerEdgeColor','k',...            %设置标记点边缘颜色为黑色
        'MarkerFaceColor','y',...            %设置标记点填充颜色为黄色
        'MarkerSize',10)                     %设置标记点的尺寸为 10
```

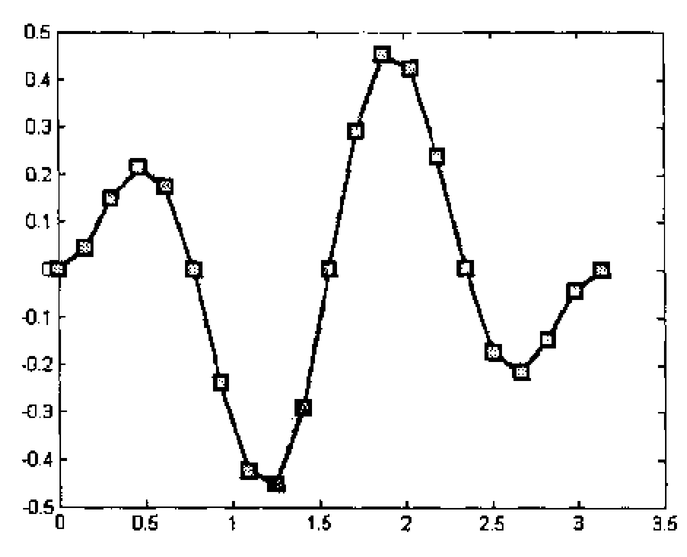


图 5-15 指定线宽、标记点颜色及尺寸

5.2 特 殊 图 形

MATLAB 支持各种类型的图形绘制，以便于将数据信息更清晰准确地表现出来，图形类型的选择通常取决于数据的特点和要表现的形式。本节将介绍 MATLAB 中一些特殊图形的绘制方法。

5.2.1 条形图

1. 二维条形图

MATLAB 中绘制条形图的基本函数为 bar 和 barh，它们的调用格式为：

• bar(*Y*) 如果 *Y* 为矢量，其每一个元素绘制一个条形；如果 *Y* 为矩阵，则 bar 函数对每一行中元素绘制的条形进行分组。

• bar(*x*, *Y*) 按 *x* 中指定的位置绘制 *Y* 中每一元素的条形。

• bar(..., width) 设置相邻条形的宽度并控制组内条形的分离，默认值为 0.8，如果指定 width=1，则组内的条形挨在一起。

• bar(..., 'style')　指定绘制条形的类型，style 有两个选项：

(1) stacked　矩阵 *Y* 中每一行绘制一个条形，条形的高度为行中元素的和，每一个条形都用多种颜色表示，颜色对应不同种类的元素并表示每行元素对总和的相对贡献；

(2) group　绘制 *n* 组条形图组，每一个条形组中有 *m* 个垂直条形，其中 *n* 对应矩阵 *Y* 的行数，*m* 对应列数，group 为 style 的默认值。

• bar(..., *LineSpec*)　用 *LineSpec* 指定的颜色绘制条形。

• *h* = bar(...)　返回图形句柄。

• barh(...)　绘制水平条形图。

• *h* = barh(...)　返回水平条形图的图形句柄。

【例 5-16】　绘制最简单的条形图。

在命令窗口创建矢量 *x*，绘制条形图（图 5-16）：

```
>> x=[4 2 6 8 1 5];
>> bar(x)
```

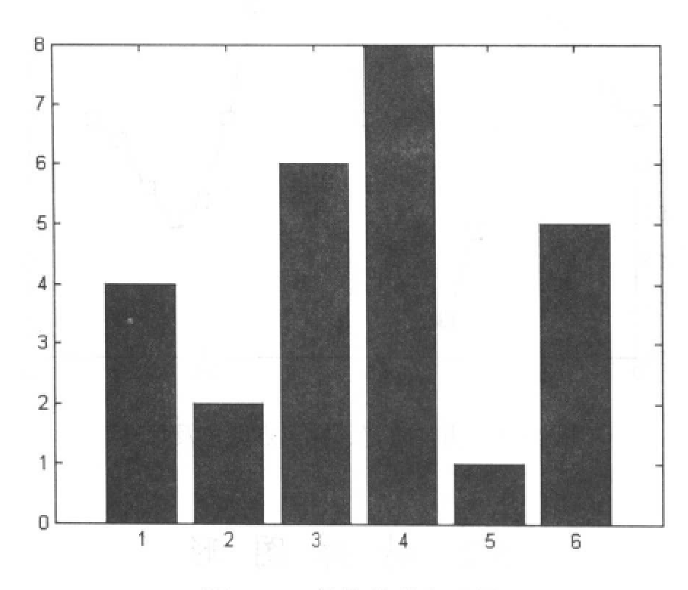


图 5-16　最简单的条形图

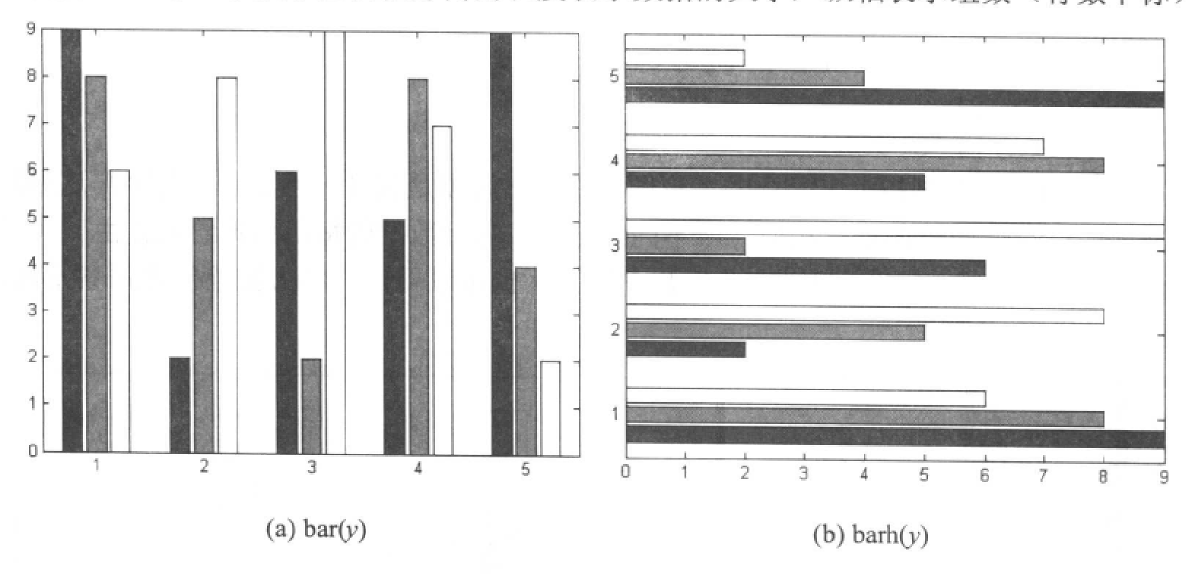
【例 5-17】　绘制矩阵的条形图。

在命令窗口创建矩阵 *y*，绘制垂直和水平条形图（图 5-17）：

```
>> y=[9 8 6;2 5 8;6 2 9;5 8 7;9 4 2]
y =
     9     8     6
     2     5     8
     6     2     9
     5     8     7
     9     4     2
>> bar(y)
>> barh(y)
```

当 *y* 是 $m \times n$ 阶的矩阵时，bar 绘制的条形图以分组或叠加的形式表现，矩阵中每一行的元素绘制在一组中，每一列元素绘制在每组中相应的位置上（即同样颜色的条形表示同一列数据）。图（a）绘制了垂直排列的条形图，按横轴方向分组，每一个条形在纵轴方向的高

度表示元素的大小，横轴表示组数（行数下标）。图（b）绘制了水平排列的条形图，按纵轴方向分组，每一个条形在横轴方向的长度表示数据的大小，纵轴表示组数（行数下标）。



(a) bar(y)　　(b) barh(y)

图 5-17　分组显示的二维条形图

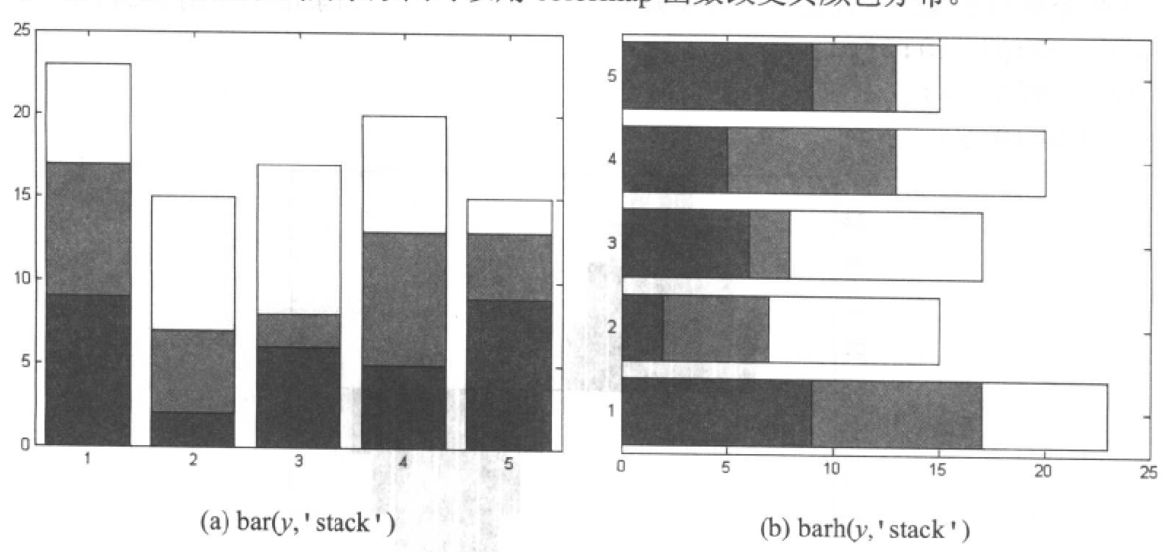
【例 5-18】　绘制叠加形式的条形图。

使用例 5-17 中创建的矩阵，绘制叠加形式的垂直和水平条形图（图 5-18）：

```
>> bar(y,'stack')
>> barh(y,'stack')
```

在叠加形式的条形图中，每一个条形由多个块组成来表示一行数据，即整个一个条形表示一行数据，同一颜色表示一列数据。在图（a）绘制的垂直条形图中，每一个条形中各块的垂直高度表示元素的大小，图（b）中各块的水平长度表示元素的大小。叠加形式的条形图通常用于表现各个分量在总体中所占的份额。

分组形式或叠加形式的条形图可以用 colormap 函数改变其颜色分布。



(a) bar(y,'stack')　　(b) barh(y,'stack')

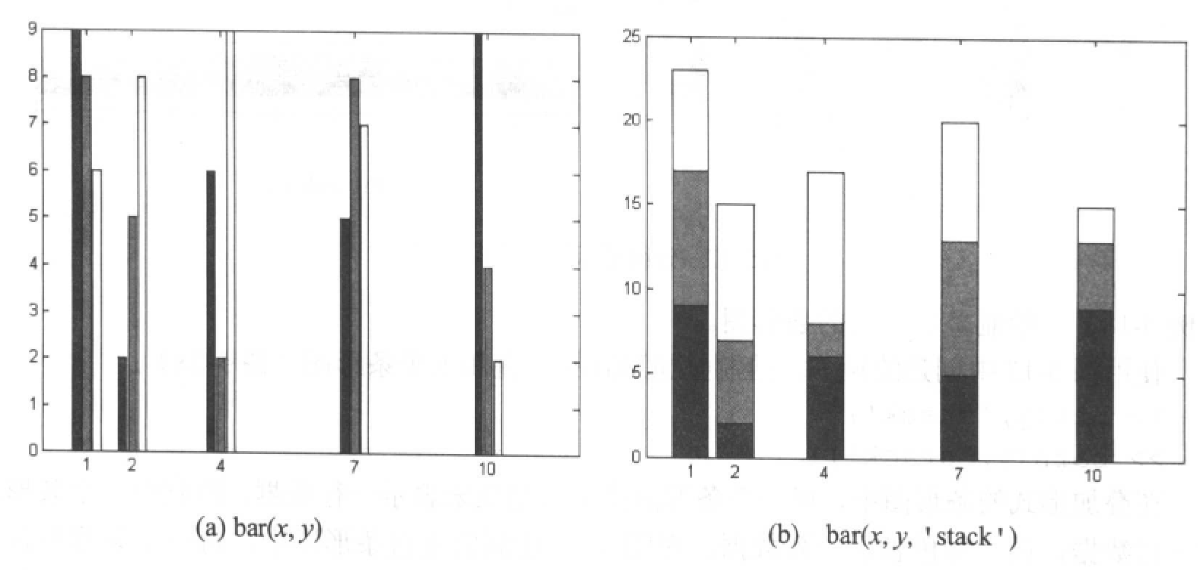
图 5-18　叠加形式显示的二维条形图

【例 5-19】　绘制指定 x 坐标的条形图。

在命令窗口创建坐标矢量 x 和矩阵 y，绘制指定 x 坐标的垂直和水平条形图（图 5-19）：

```
>> x=[1 2 4 7 10]
x =
     1     2     4     7    10
>> y=[9 8 6;2 5 8;6 2 9;5 8 7;9 4 2];
>> bar(x,y)
>> bar(x,y,'stack')
```

使用 bar(*x*, *y*)绘制指定 *x* 坐标的条形图，其中 *x* 必须是矢量。当 *y* 是矩阵时，这种调用方式与使用 bar(*y*)的绘图形式相似，所不同的是代表每行数据的每组条形绘制在由 *x* 矢量指定的横轴位置上（而不是 bar(*y*)中的行数下标），也可以用同样的方法绘制水平条形图，*x* 确定的是纵轴的位置。



(a) bar(*x*, *y*)　　(b) bar(*x*, *y*, 'stack')

图 5-19　指定 *x* 坐标的二维条形图

【例 5-20】　绘制两个矢量的条形图。

在命令窗口创建坐标矢量 *x* 及函数矢量 *y*，绘制两个矢量的条形图（图 5-20）：

```
>> x=0:pi/10:2*pi;
>> y=sin(x);
>> bar(x,y)
```

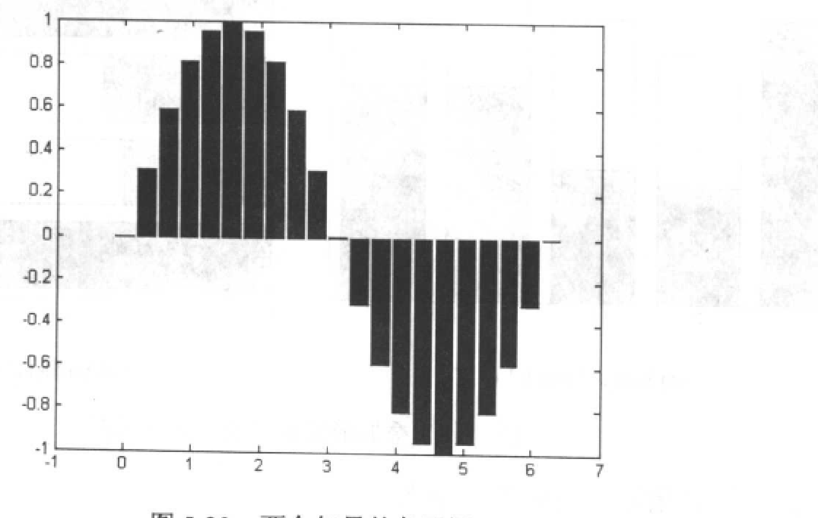


图 5-20　两个矢量的条形图

如果 x 和 y 为同样尺寸的矢量，对应于每一个 x 元素绘制一个条形，条形的高度表示矢量 y 元素的大小，这种绘图更接近二维曲线绘图形式（可以把二维曲线图看成是这种条形图的包络线）。

在 bar 函数中，可以使用与 plot 命令相同的方法改变图形的颜色，如 bar(x, y, 'r')可以绘制出红色的条形图。

2. 三维条形图

三维条形图的绘制函数为 bar3 和 bar3h，其使用方法与 bar 和 barh 基本相同，其调用格式为：

- bar3(*Y*) 绘制三维条形图，如果 *Y* 为矢量，其每一个元素绘制一个条形，如果 *Y* 为矩阵，则 bar3 函数对每一行中元素绘制的条形进行分组。
- bar3(*x*, *Y*) 按 *x* 中指定的位置绘制 *Y* 中每一元素的条形。
- bar3(..., width) 设置相邻条形的宽度并控制组内条形的分离，默认值为 0.8，如果指定 width=1，则组内的条形挨在一起。
- bar3(..., 'style') 指定绘制条形的类型，style 有 3 个选项：

(1) detached 在 *x* 方向用单独的条形块绘制 Y 中的每一行元素，detached 为 style 的默认值；

(2) stacked 矩阵 *Y* 中每一行绘制一个条形，条形的高度为行中元素的和，每一个条形都用多种颜色表示，颜色对应不同种类的元素并表示每行元素对总和的相对贡献；

(3) group 绘制 *n* 组条形图组，每一个条形组中有 *m* 个垂直条形，其中 *n* 对应矩阵 *Y* 的行数，*m* 对应列数。

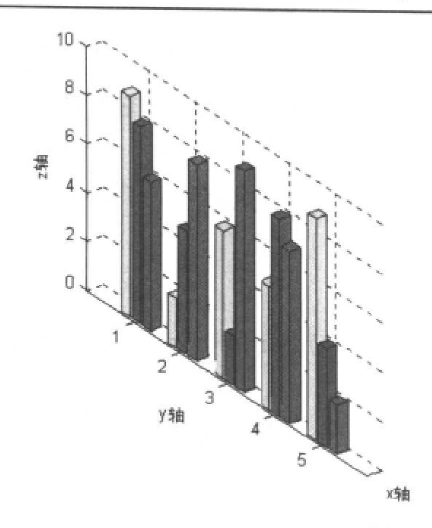
- bar3(..., *LineSpec*) 用 *LineSpec* 指定的颜色绘制条形。
- *h* = bar3(...) 返回三维条形图的图形句柄。
- bar3h(...) 绘制三维水平条形图。
- *h* = bar3h(...) 返回三维水平条形图的图形句柄。

三维条形图实质上是将 $m\times n$ 阶的矩阵绘制成分布在三维空间中的柱体。三维条形图有两种显示形式：分组形式和分列形式。三维条形图的分组显示与二维条形图的分组显示形式基本相同，只是将平面的条形变为立体的柱形，在 x 轴方向没有意义。分列形式的三维条形图更接近真正的三维显示，在 x 轴方向表示列序号，在 y 轴方向表示行序号，每个柱体的高度由矩阵元素的大小确定（如果显示为水平形式，则在 z 轴方向表示行序号，在 y 轴方向表示矩阵元素的大小）。

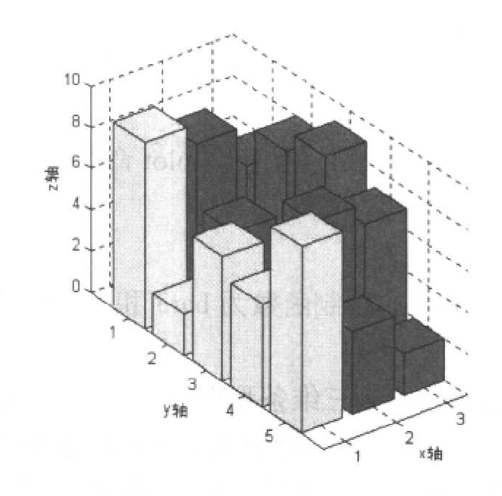
【例 5-21】 分别按分组和分列绘制三维条形图。

创建矩阵 y，分别按分组和分列绘制三维条形图（图 5-21）：

```
>> y=[9 8 6;2 5 8;6 2 9;5 8 7;9 4 2];
>> bar3(y,'group')
>> bar3(y)
>> xlabel('x轴');ylabel('y轴');zlabel('z轴')
```



(a) 分组形式（bar3(y,'group')）



(b) 分列形式（bar3(y)）

图 5-21　三维条形图

3. 条形图中的图形叠加

通过在相同的位置创建一个与原来条形图中的坐标轴相对独立的新坐标轴，可以实现在条形图中的图形叠加。

【例 5-22】 有两组生物医学的实验数据，一组表示物质成分（TCE），一组表示温度（temp），数据是在 35 天中每隔 5 天采样取得的，将物质成分和温度与时间的关系画在一张图中。

在命令窗口输入实验数据，将两组数据绘制在一个图中（图 5-22）：

```
TCE=[515 420 370 250 135 120 60 20];          % 实验数据
temp=[29 23 27 25 20 23 23 27];
days=0:5:35;                                  % 采样天数
bar(days,temp,'c')                            % 温度与时间的条形图
xlabel('Day')
ylabel('Temperature (^{o}C)')
h1=gca;                                       % 获取当前坐标轴对象句柄
% 建立与 h1 位置相同、句柄为 h2 的新坐标轴对象
h2=axes('Position',get(h1,'Position'));
% 在以 h2 为句柄的坐标轴对象上绘制物质成分与时间的关系曲线
plot(days,TCE,'LineWidth',3)
% 设置句柄为 h2 的坐标轴对象的 Y 轴位置为左侧
set(h2,'YAxisLocation','right','Color','none','XTickLabel',[])
%设置句柄为 h2 的坐标轴对象的 X 轴的范围与句柄为 h1 的坐标轴对象的 X 轴的范围相同
set(h2,'XLim',get(h1,'XLim'),'Layer','top')
% 在坐标为[11,380]的位置以旋转-55°的方向书写 Concentration
text(11,380,'Concentration','Rotation',-55,'FontSize',16)
ylabel('TCE Concentration (PPM)')
title('Bioremediation','FontSize',16)
```

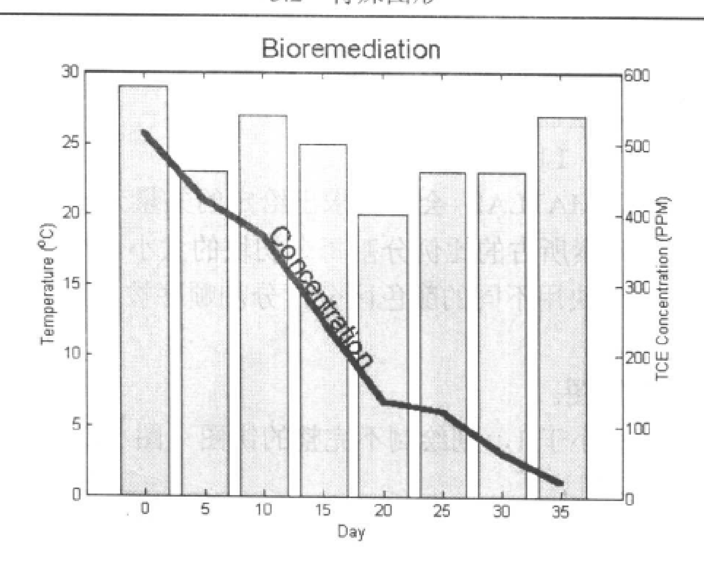


图 5-22 在条形图上叠加线图的实例

本例只是为了显示条形图与线图叠加的方法和效果，例中一些语句涉及到图形句柄的内容，请参阅 6.1。

5.2.2 饼图

饼图用于表示矢量或矩阵中各元素所占的比例，MATLAB 中 pie 和 pie3 函数提供了平面饼图和三维饼图的绘制功能，它们的调用格式为：

- pie(*X*) 使用 *X* 中的数据绘制饼图，*X* 中的每一元素用饼图中的一个扇区表示。
- pie(*X*, *explode*) 将一些扇区从饼图中分离出来，*explode* 为一个与 *X* 尺寸相同的矩阵，其非零元素所对应的 *X* 矩阵中的元素从饼图中分离出来。
- *h* = pie(...) 返回图形对象的句柄。

【例 5-23】 绘制不分离饼图。

在命令窗口创建矢量 *x*，绘制饼图（图 5-23）：

```
>> x=[5.5 74.7 44.5 33.2 46.6];
>> pie(x)
```

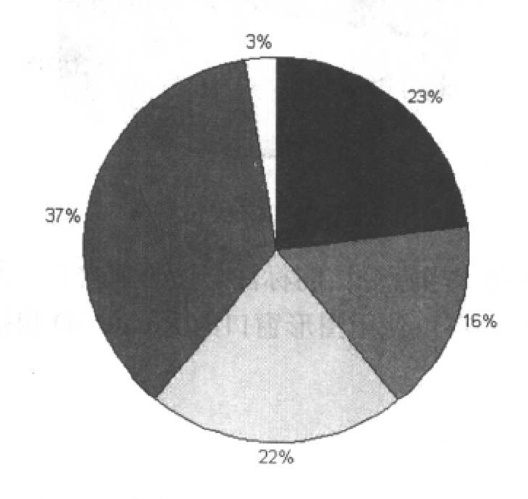


图 5-23 不分离的饼图

【例 5-24】　绘制分离饼图。

使用例 5-23 的矢量 *x*，绘制分离饼图（图 5-24）：

```
>> pie(x,[0 0 0 0 1])
```

在计算饼图的比例时，MATLAB 会自动求出给定的矢量元素之和，然后再分别算出各元素所占的比例，按照各元素所占的比例分割每个切块的大小，并以百分比的形式自动标注在相应的切块旁边，每个切块用不同的颜色区分，分割顺序按元素下标从 90°角位置逆时针旋转。

【例 5-25】　绘制不完整饼图。

如果 *x* 的全部元素之和小于 1，则绘制不完整的饼图（图 5-25）：

```
>> x=[0.2 0.3 0.4];
>> pie(x)
```

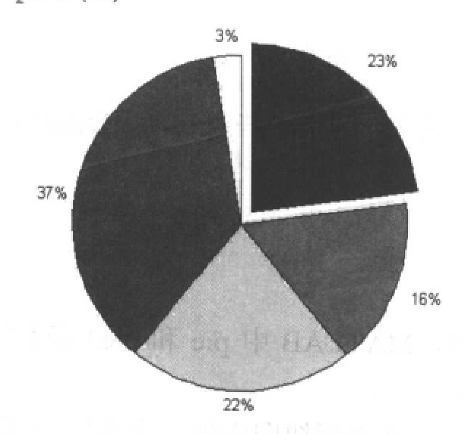


图 5-24　带分离切块的饼图

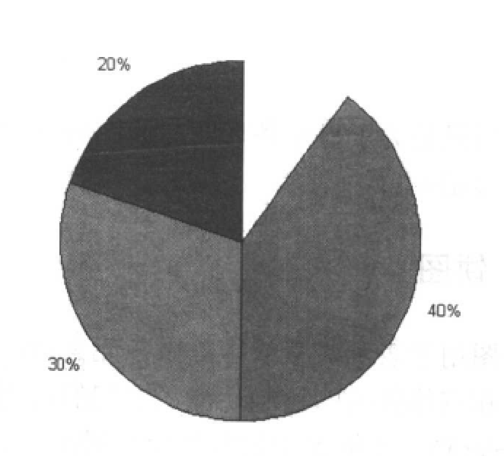


图 5-25　不完整的饼图

【例 5-26】　绘制三维饼图。

三维饼图是有一定厚度的饼图，绘图函数为 pie3，调用方法与二维饼图相同（图 5-26）。

```
>> pie3([1 2 4 3 5],[0 1 0 1 0])
```

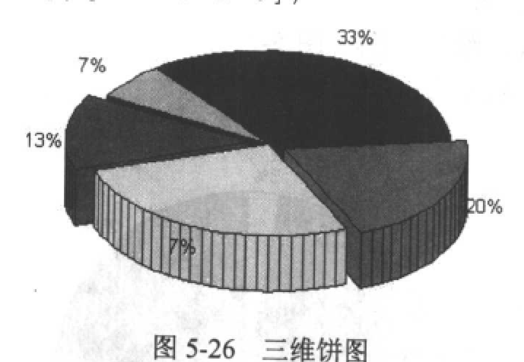


图 5-26　三维饼图

注意，在绘制三维饼图时，当百分比的标注与图形重叠时，显示的数值可能不全（如图 5-26 的百分比总和为 80%），可以使用图形窗口的 Rotate 3D 快捷按钮改变视图，以得到正确的显示。

5.2.3　其他图形

MATLAB 6.1 提供了 20 余种特殊图形的绘制方法，由于篇幅的限制，这里不做详细介

绍，只对一些常用的函数举例说明。

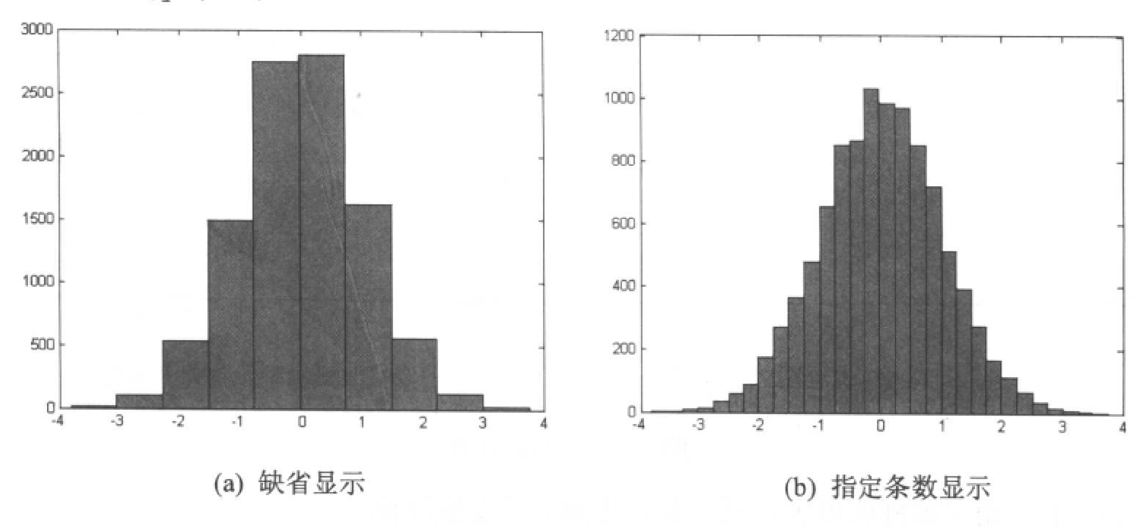
1. 直方图

直方图（hist）与条形图（bar）从表面上看很相似，但实质上是不同的，条形图只是简单地用条状图形将数据点表现出来，而直方图则是一种统计运算的结果，它的横轴是数据的幅度，纵轴是对应于各个幅度数据出现的次数，直方图纵坐标没有负数。

【例 5-27】 绘制直方图。

使用 randn 函数产生 10 000×1 阶的矢量，绘制其直方图（图 5-27）：

```
>> yn=randn(10000,1);
>> hist(yn)
>> hist(yn,20)
```



(a) 缺省显示 (b) 指定条数显示

图 5-27 矢量直方图

在图 5-27（a）中，缺省状态下只绘制 10 个条形，而图 5-27（b）中通过指定参数绘制了 20 个条形。

【例 5-28】 绘制矩阵的直方图。

使用 randn 函数产生 10 000×3 阶的矩阵，绘制其直方图（图 5-28）：

```
>> y=randn(10000,3);
>> hist(y)
```

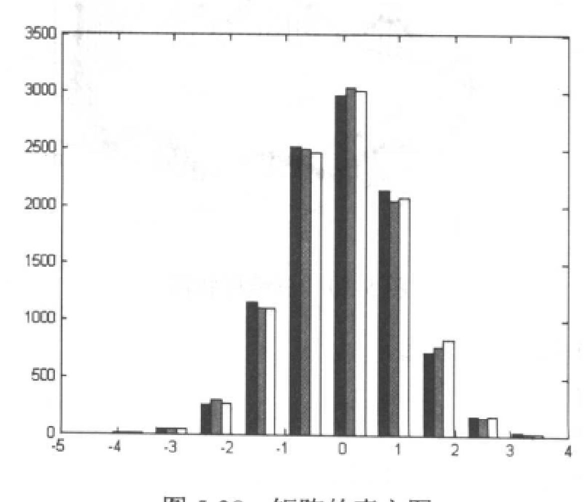


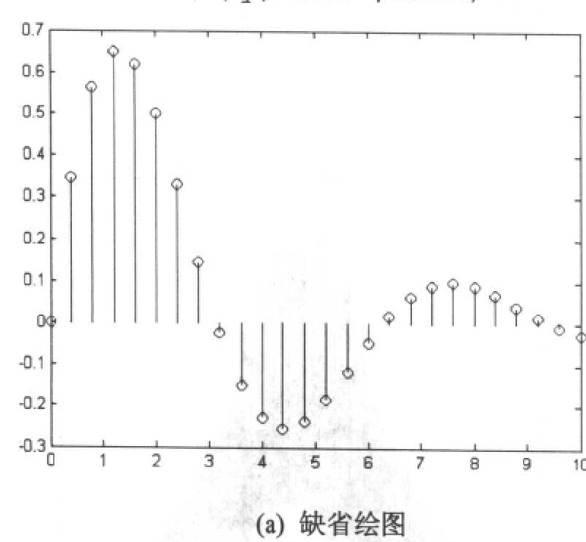
图 5-28 矩阵的直方图

2. 用杆状图表现离散数据

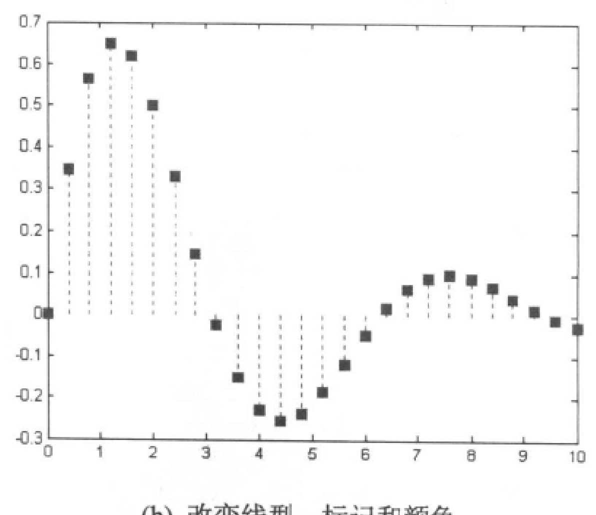
【例 5-29】　绘制二维杆状图。

在命令窗口创建 x 和 y，绘制二维杆状图（图 5-29）。

```
>> x=0:0.2:10;
>> y=exp(-0.3*x).*sin(x);
>> stem(x,y)
>> stem(x,y,':sr',fill)                        %改变线型、标记点形状和颜色
```



(a) 缺省绘图



(b) 改变线型、标记和颜色

图 5-29　二维杆状图

【例 5-30】　用三维杆状图表现复平面快速傅里叶变换计算。

在命令窗口创建角度矢量 th，并创建函数 x、y 和 f，绘制三维杆状图（图 5-30）。

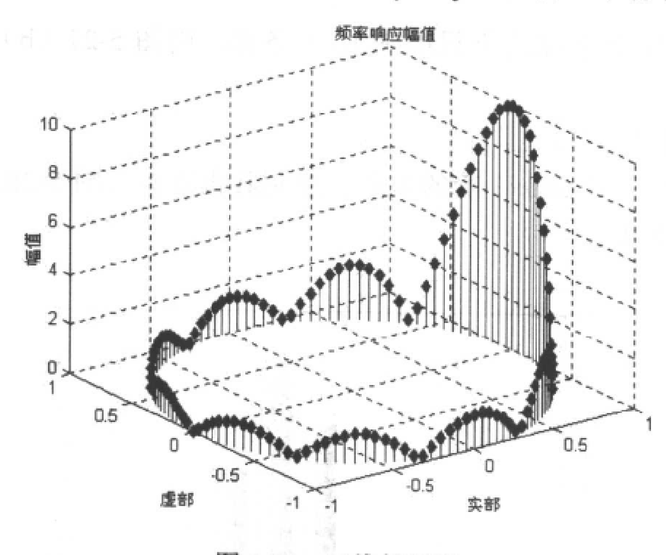


图 5-30　三维杆状图

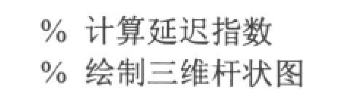
```
>> th = (0:127)/128*2*pi;
>> x = cos(th);
>> y = sin(th);
>> f = abs(fft(ones(10,1),128));
>> stem3(x,y,f','d','fill')
```

```
>> xlabel('实部')
>> ylabel('虚部')
>> zlabel('幅值')
>> title('频率响应幅值')
```

【例 5-31】 用三维杆状图与其他图形的叠加表现拉普拉斯变换基函数。

在命令窗口创建时间矢量 t，并创建函数 y，绘制三维杆状图（图 5-31）。

```
>> t=0:0.1:10;                                % 时间范围
>> s=0.1+i;
>> y=exp(-s*t);                               % 计算延迟指数
>> stem3(real(y),imag(y),t,'m')               % 绘制三维杆状图
>> hold on                                    % 保持绘图
>> hline=plot3(real(y),imag(y),t, 'k')        % 返回三维曲线图的句柄
>> hold off
>> set(hline,'LineWidth',3)                   % 设置线宽
>> xlabel('实 部')
>> ylabel('虚 部')
>> zlabel('幅 值')
```



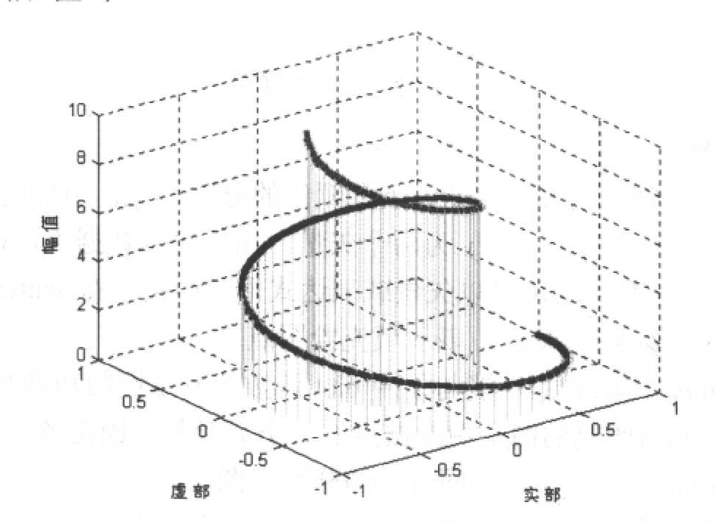


图 5-31 三维杆状图与其他图形的叠加

3. 阶梯图

【例 5-32】 绘制函数 $f = e^{-\alpha t \sin(\beta t)}$ 阶梯图。

在命令窗口创建角度变量 *alpha*、*beta*、时间矢量 t，并创建函数 f，绘制阶梯图（图 5-32）：

```
>> alpha=0.01;
>> beta=0.5;
>> t=0:10;
>> f=exp(-alpha*t).*sin(beta*t);
>> stairs(t,f)
>> hold on
>> plot(t,f,':*')
>> hold off
>> label='函数 e^{-(\alpha*t)} sin\beta*t 的阶梯图';
```

```
>> text(0.5,-0.2,label,'FontSize',14)
>> xlabel('t = 0:10','FontSize',14)
>> axis([0 10 -1.2 1.2])
```

在图 5-32 中，实线表示的是阶梯图，叠加的短虚线加标记点图为了说明数据点在阶梯图中的位置（每一级阶梯的起始点为数据点）。

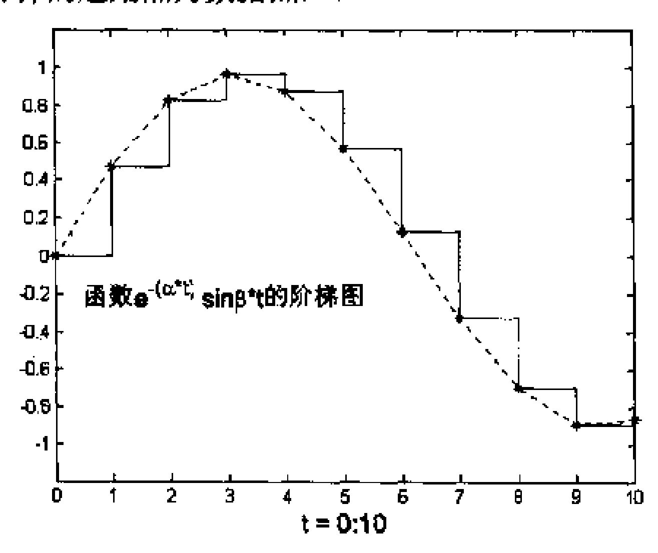


图 5-32　绘制阶梯图

4. 彩色分散点图

scatter 函数用来绘制可以改变点的大小和颜色的分散点图，函数的调用格式为：

- scatter(*x*, *y*, *s*, *c*)　其中：*x* 和 *y* 为两个矢量，用于定位数据点，*s* 为绘图点的大小，*c* 为绘图所使用的色彩，*s* 和 *c* 均可以以矢量或表达式形式给出，在 scatter 函数的前 4 个参数之后还可以增加第 5 个参数'filled'，表示填充绘图点。

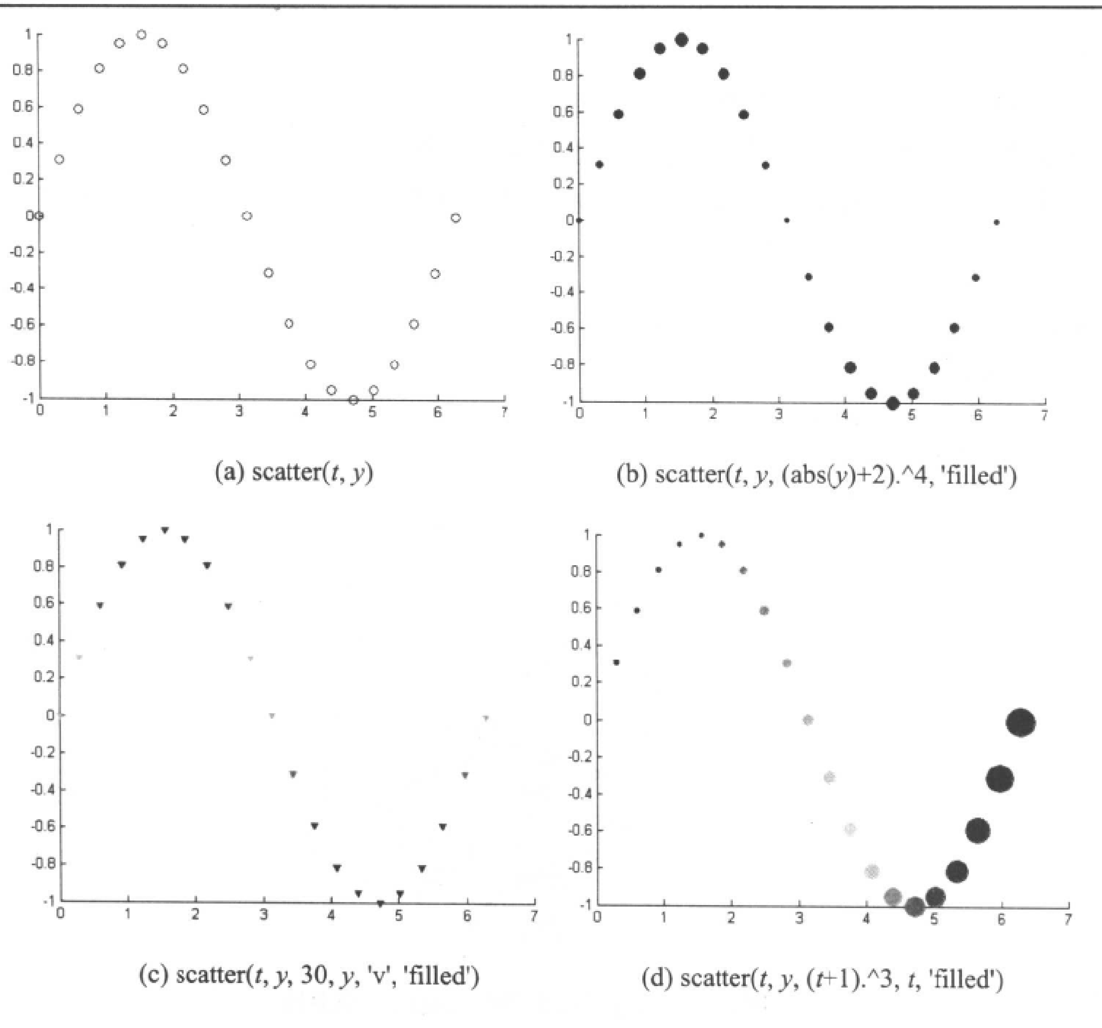
如果不指定 *s* 和 *c*，该函数的功能与使用 plot 函数绘制点图的功能相同，如果只指定了 *s* 或 *c* 中的一个参数，则所绘制的图形中数据点的大小或颜色是固定的。

【例 5-33】　使用 scatter 函数绘制不同效果的分散点图。

在命令窗口创建矢量 *t*，并创建函数 *y*，绘制分散点图（图 5-33）：

```
>> t=0:pi/10:2*pi;
>> y=sin(t);
>> scatter(t,y)
>> scatter(t,y,(abs(y)+2).^4,'filled')
>> scatter(t,y,30,y,'v','filled')
>> scatter(t,y,(t+1).^3,t,'filled')
```

从图中可以看到(a)的结果使用 plot 函数也可以实现，而 scatter 与 plot 的最大差别在于 scatter 命令可以绘制变尺寸、变颜色的分散点图。



(a) scatter(*t*, *y*)　　(b) scatter(*t*, *y*, (abs(*y*)+2).^4, 'filled')

(c) scatter(*t*, *y*, 30, *y*, 'v', 'filled')　　(d) scatter(*t*, *y*, (*t*+1).^3, *t*, 'filled')

图 5-33 彩色分散点图

5.3 三 维 图 形

MATLAB 提供了大量三维图形的表现函数，通过这些函数可以绘制三维曲线图、网格图(Mesh Plot)、表面图(Shaded Surface Plot)、伪彩图(Pseudcolor Plot)和等高线图(Contour)。此外，MATLAB 还提供了控制颜色、光线、视角等绘图效果的函数和命令，从而使三维图形的表现更加灵活自如。

5.3.1 三维曲线图

函数 plot3 用于绘制三维曲线图，其调用格式为：

- plot3(*X*1, *Y*1, *Z*1 ,...)　绘制由一系列数据点确定的三维曲线图，其中 *X*1、*Y*1 和 *Z*1 为决定数据点三维坐标的矢量或矩阵。
- plot3(*X*1, *Y*1, *Z*1, *LineSpec*, ...)　按照 *LineSpec* 所定义的线型、标记点和颜色绘图，具体使用方法参见 5.1.3。
- plot3(..., '*PropertyName*', *PropertyValue*, ...)　为所有由 plot3 函数创建的图形对象设置

属性值。

- h = plot3(...)　返回一个图形函数句柄。

【例 5-34】　绘制 x、y 和 z 均为矢量时的三维曲线图。

在命令窗口创建自变量矢量 t，并创建函数 x、y 和 z，绘制三维曲线图（图 5-34）：

```
>> t=0:pi/50:10*pi;
>> x=exp(-t/15).*sin(2*t);
>> y=exp(-t/15).*cos(2*t);
>> z=t;
>> plot3(x,y,t)
>> axis square; grid on
```

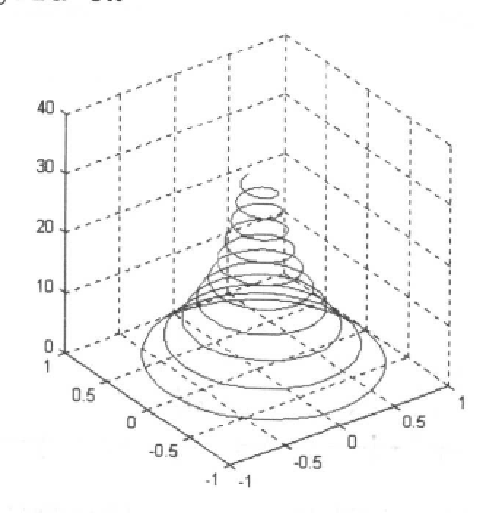


图 5-34　x、y 和 z 均为矢量时的三维曲线

【例 5-35】　绘制 x、y 和 z 均为矩阵时的三维曲线图。

在命令窗口产生供三维绘图的网格数据[X,Y]，并创建函数 Z，绘制三维曲线图（图 5-35）。

```
>> [X,Y]=meshgrid([-2:0.1:2]);          %产生供三维绘图的网格矩阵 x、y
>> Z=X.*exp(-X.^2-Y.^2);
>> plot3(X,Y,Z)
>> grid on
```

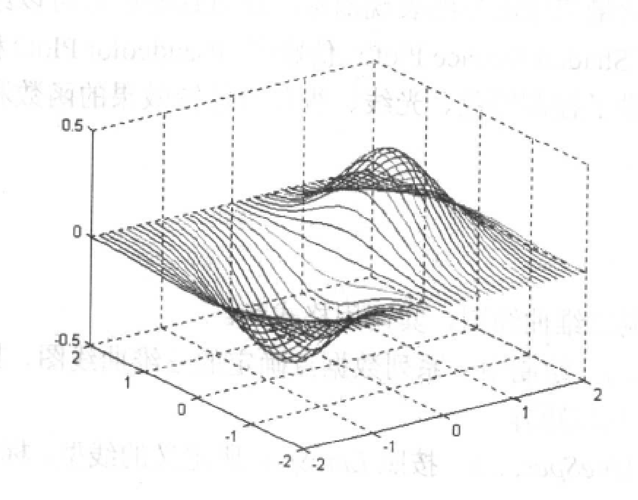


图 5-35　x、y 和 z 均为矩阵时的三维曲线

从图中可以看到，在这种图形的绘图过程中自动将 z 矩阵的每列数据作为一条曲线，不

同列的数据用不同的颜色表示，曲线的颜色按照 7 种预设的颜色顺序循环。

在调用 plot3 函数时，与 plot 函数一样也可以设置线型、标记点的形状和曲线的颜色等参数。

【例 5-36】 按指定的线型、颜色及标记点绘制三维曲线图。

在命令窗口矢量 t，指定线型、颜色及标记点，绘制三维曲线图（图 5-36）：

```
>> t=0:pi/20:5*pi;
>> plot3(sin(t),cos(t),t,':*r')
```

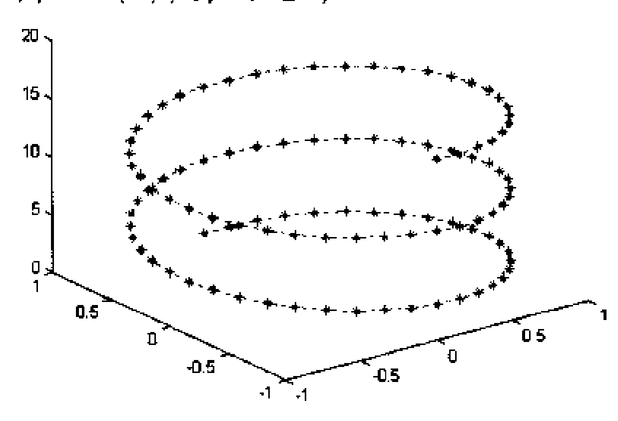


图 5-36 指定线型、颜色及标记点的三维曲线

5.3.2 三维绘图数据的产生

MATLAB 提供了一组用于生成三维绘图数据的函数，这些函数包括：

(1) peaks

用于创建双峰函数和用双峰函数绘图，其基本调用格式为：

- [x, y, z]=peaks(n) 分别创建 x、y、z 三个均为 $n\times n$ 阶的方阵，其中 x 的每一列的元素都相同，每一行的元素均为在[−3,3] 区间内的 n 等分，y 的行列元素与 x 刚好相反，z 为 x 和 y 的函数，$z=3(1-x)^2e^{-x^2-(y+1)^2}-10\left(\frac{x}{5}-x^3-y^5\right)e^{-x^2-y^2}-\frac{1}{3}e^{-(x+1)^2-y^2}$，$n$ 的缺省值为 49。

- peaks(n) 直接使用 peaks 函数所创建的 x、y、z 三个矩阵绘制表面图，相当于执行了[x, y, z]=peaks(n)和 surf(x, y, z)两个函数。

(2) meshgrid

按指定方式生成网格矩阵，其基本调用格式为：

- [X, Y, Z]= meshgrid(x, y, z) 分别创建 X、Y、Z 三个 $m\times n\times k$ 阶的矩阵，矩阵的阶数由 x、y、z 三个矢量的长度 m、n、k 确定，X、Y、Z 三个矩阵表示了三维空间的网格，如果省略了参数 Z，则创建二维网格。

5.3.3 网格图

mesh(z)语句绘制矩阵 z 元素的三维网格图，即不着色的表面图。它所产生的网格形表面由生成 x-y 平面的网格对应的 z 坐标定义，图形由邻近点用直线连接而成。

mesh 函数的调用格式为：

• mesh(z)　绘制分别以 $m \times n$ 阶矩阵 z 的行数和列数为 x 和 y 坐标（[1:m], [1:n]），以 z 的对应元素值为 z 坐标的三维网格图。

• mesh(x, y, z)　绘制分别以矩阵 x、y、z 的元素值为坐标的三维网格图，x、y、z 必须为同阶矩阵。

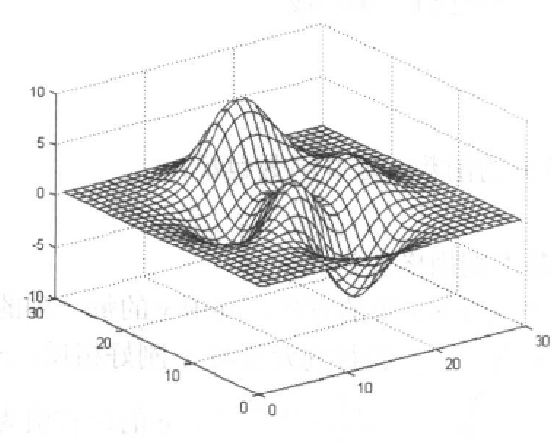
除基本的网格图绘制函数 mesh 以外，MATLAB 还提供了另外两个绘制网格图的函数：meshc 和 meshz，其中 meshc 将网格图与等高线图一起绘制，meshz 在绘制网格图的同时绘制零基准平面图。

【例 5-37】　绘制各种网格图。

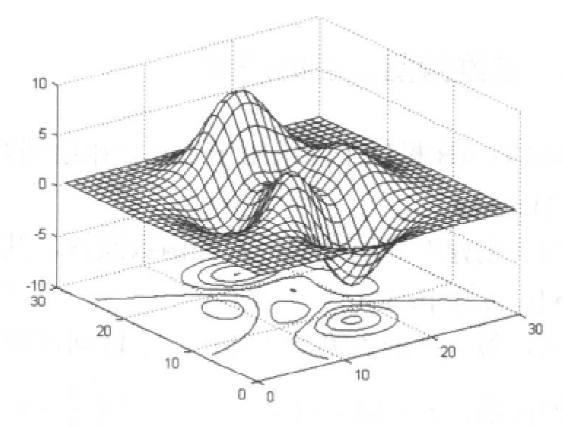
使用 mesh、meshc、meshz 和 waterfall 等函数绘制网格图（图 5-37）。

```
>> mesh(peaks(30));                      %最基本的网格图
>> meshc(peaks(30));                     %绘制带等高线的网格图
>> p=peaks(30);
>> p(20:23,9:15)=NaN*ones(4,7);          %将一部分网格设置为非数
>> meshz(p);                             %带剪孔的垂帘网格图
>> waterfall(peaks(30));                 %绘制瀑布水线图，为网格图的一个变种
```

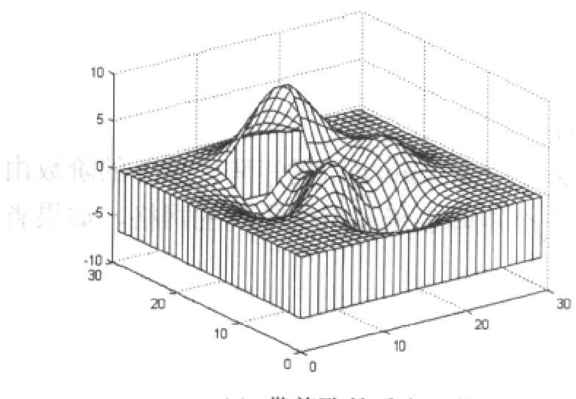
在 MATLAB 中，常数 NaN 定义为非数，绘图中遇到非数的数据点时不绘制该点，因此，可以利用 NaN 表现出剪孔的效果。



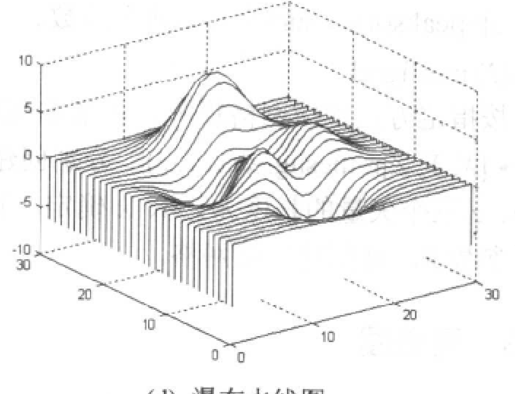
(a) 最基本的网格图



(b) 带等高线的网格图



(c) 带剪孔的垂帘网格图



(d) 瀑布水线图

图 5-37　各种网格图

5.3.4 着色表面图

MATLAB 提供了三个绘制着色表面图的命令：surf、surfc 和 surfl，其中 surf 为绘制着色表面图的基本命令，surfc 为绘制带等高线的着色表面图，surfl 可以控制表面图的光照效果。

函数 surf 的调用格式为：

- surf(*Z*) 绘制分别以 $m\times n$ 阶矩阵 *Z* 的行数和列数为 *x* 和 *y* 坐标（[1:*m*], [1:*n*]），以 *Z* 的对应元素值为 *Z* 坐标的三维表面图。
- surf(*X*, *Y*, *Z*) 绘制分别以矩阵 *X*、*Y*、*Z* 的元素值为坐标的三维表面图，*X*、*Y*、*Z* 必须为同阶矩阵。
- surf(*X*, *Y*, *Z*, *C*) 用 *C* 定义的颜色绘制三维表面图。
- surf(...,'*PropertyName*', *PropertyValue*) 定义三维表面图属性。
- surfc(...) 绘制带等高线的三维表面图。
- *h* = surf(...)和 *h* = surfc(...) 返回表面图的图形对象句柄。

【例 5-38】 绘制椭圆表面图。

生成数据并使用 surf 函数绘制椭圆表面图（图 5-38）：

```
>> x=-1.5:0.3:1.5;
>> y=-1:0.2:1;
>> [x,y]=meshgrid(x,y);
>> z=sqrt(4-x.^2/9-y.^2/4);
>> surf(x,y,z)
```

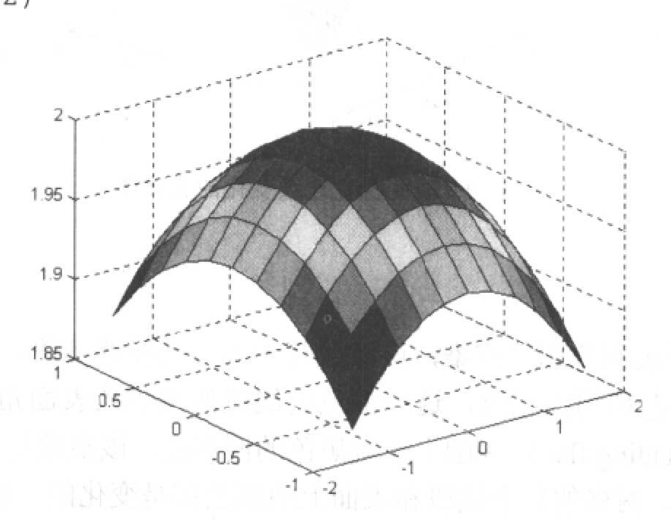


图 5-38 椭圆着色表面图

【例 5-39】 绘制带光照效果的表面图。

使用例 5-38 的数据绘制带光照效果的椭圆表面图，利用 colormap 函数变化色彩（图 5-39）：

```
>> surfl(x,y,z)
>> colormap(hot)
```

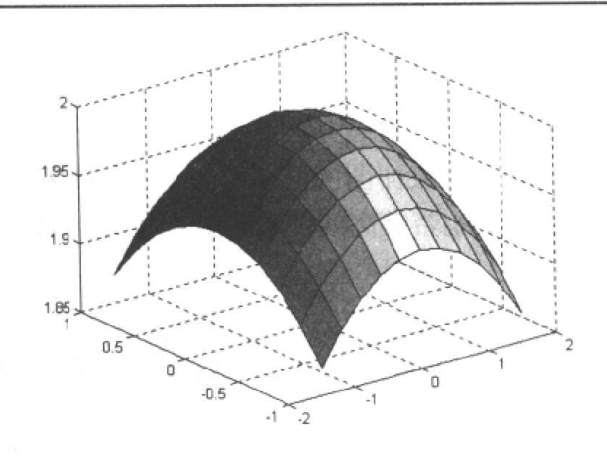


图 5-39　光照下的椭圆着色表面

【例 5-40】　绘制表面颜色连续变化的表面图。

在例 5-39 的基础上，通过调用表面色彩渲染函数 shading，控制表面颜色连续变化（图 5-40）：

```
>> shading interp
```

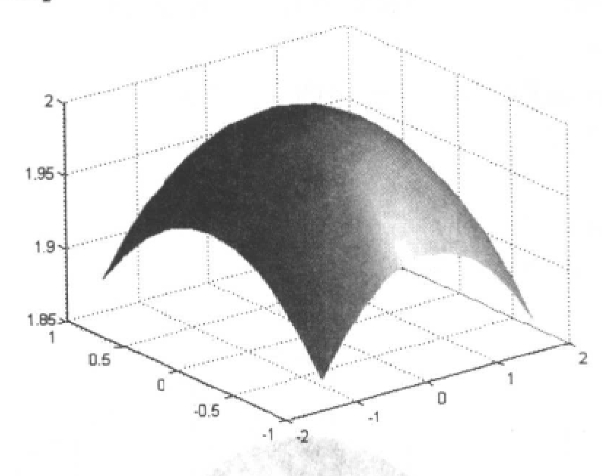


图 5-40　表面颜色连续变化的表面图

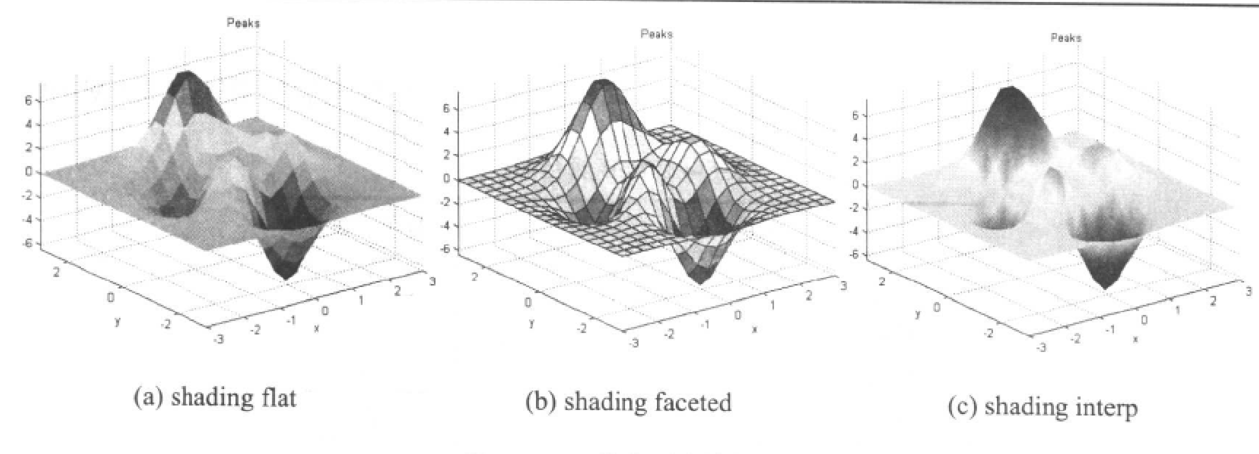
5.3.5　表面渲染

shading 函数用于表面色彩的渲染，该函数有 3 个可选择的参数：(1) shading flat。网格的每个线段和表面都有相同的颜色，这个颜色由线段的端点或表面角上的色彩值决定；(2) shading faceted。在 shading flat 的基础上绘制黑色的网格线，该参数是 shading 函数的缺省状态；(3) shading interp。网格的每个线段和表面上的颜色都是变化的，该参数通过对线段或表面上的颜色进行插值，使得整个表面上的颜色看上去是连续变化的。

【例 5-41】　比较表面渲染的效果。

生成双峰函数，绘制表面图，比较三种表面渲染效果（图 5-41）：

```
>> surf(peaks(30))
>> shading flat
>> shading faceted
>> shading interp
```



(a) shading flat (b) shading faceted (c) shading interp

图 5-41 3 种表面渲染的效果

5.3.6 表面图形的透明处理

在缺省情况下，MATLAB 总是将网格图后面的线条隐藏起来，即便没有颜色的表面，也看不到后面的网格，所以网格图看起来是不透明的，利用下面的命令可以使图形透明：

- hidden off hidden 是一个开关型命令，与之对应的为 hidden on。

【例 5-42】 比较透明处理后网格图的变化。

生成网格点，绘制网格图，比较透明处理前后的网格图（图 5-42）：

```
>> [X,Y]=meshgrid(-8:.5:8);
>> R=sqrt(X.^2+Y.^2)+eps;
>> Z=sin(R)./R;
>> mesh(Z)
>> colormap([0 0 1])              %将色彩图设置为单色（缺省状态无透明处理）
>> hidden off                     %使用透明处理
```

(a) hidden on（缺省） (b) hidden off

图 5-42 透明处理前后的效果对比

5.3.7 表面图形的颜色映射

colormap 函数用于按照预定的色彩索引映射色彩图。colormap(*MAP*)是将当前图形窗口

的色彩图设定为 *MAP*。

表 5-4 列出了典型的 RGB 颜色表，表 5-5 列出了 MATLAB 预设的色彩映射表（MAP）。

表 5-4　RGB 颜色表

颜色名称	红色（R）	绿色（G）	蓝色（B）
黑色	0	0	0
白色	1	1	1
红色	1	0	0
绿色	0	1	0
兰色	0	0	1
黄色	1	1	0
粉红色	1	0	1
淡兰色	0	1	1
灰色	0.5	0.5	0.5
深红色	0.5	0	0
紫铜色	1	0.62	0.40
碧绿色	0.49	1	0.83

表 5-5　色彩映射表（MAP）

参数	色彩图	参数	色彩图
bone	连续变化的兰色调灰度图	spring	连续变化的黄粉色调色彩图
copper	连续变化的纯铜色彩图	summer	连续变化的黄绿色调色彩图
gray	连续变化的灰度图	autumn	连续变化的黄红色调色彩图
hot	连续变化的暖色彩图	winter	连续变化的兰绿色调色彩图
pink	连续变化的粉红色彩图	flag	交错变化的四色色彩图
cool	连续变化的冷色调图	prism	交错变化的光谱色彩图
hsv	连续变化的饱和色彩图	colorcube	交错变化的浓淡色彩图
jet	连续变化的变异饱和色彩图	lines	交错变化的线图色彩图

【例 5-43】　利用色彩映射表表现特殊效果。

生成数据点，绘制表面图，利用色彩映射表表现特殊效果（图 5-43）：

```
>> t=0:pi/12:3*pi;
>> r=abs(exp(-0.25*t).*sin(t));
>> [x,y,z]=cylinder(r,30);
>> surfl(x,y,z)                                    %使用连续变化的粉红色彩图
>> colormap(pink)
>> shading interp
```

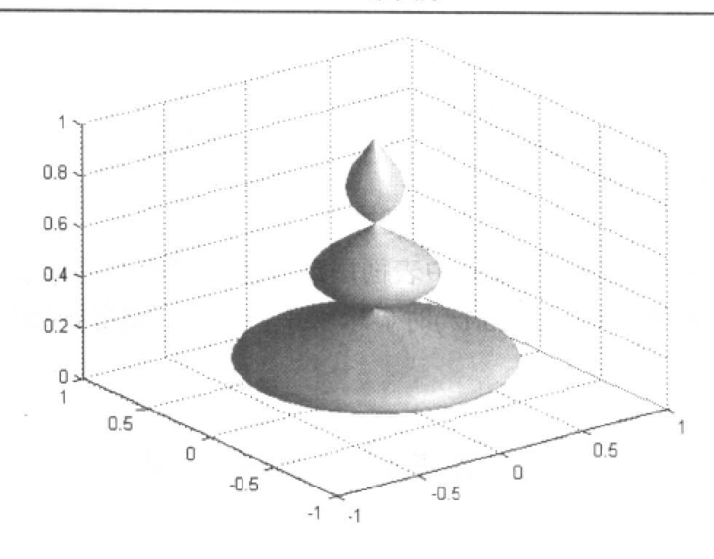


图 5-43　利用色彩映射表表现特殊效果

5.3.8　切片图

对于定义在表面、柱面和球面上的三元函数都可以借助色彩来实现四维表现，下面介绍定义在一般 *x*-*y*-*z* 坐标上的四维可视化函数 slice。

为实现三元函数 $v=f(x, y, z)$的可视化表现，MATLAB 提供了一个绘制四维切片图函数 slice，其调用格式为：

• slice(*X*, *Y*, *Z*, *v*, *xi*, *yi*, *zi*)　其中 *X*、*Y*、*Z* 为使用 meshgrid 函数生成的三维网格坐标矩阵，*v* 为所绘制图形的函数，*xi*、*yi*、*zi* 为切片位置。

【例 5-44】　绘制函数 $v=xe^{-(x^2+y^2+z^2)}$ 的四维表现图。

生成数据点，绘制切片图（图 5-44）：

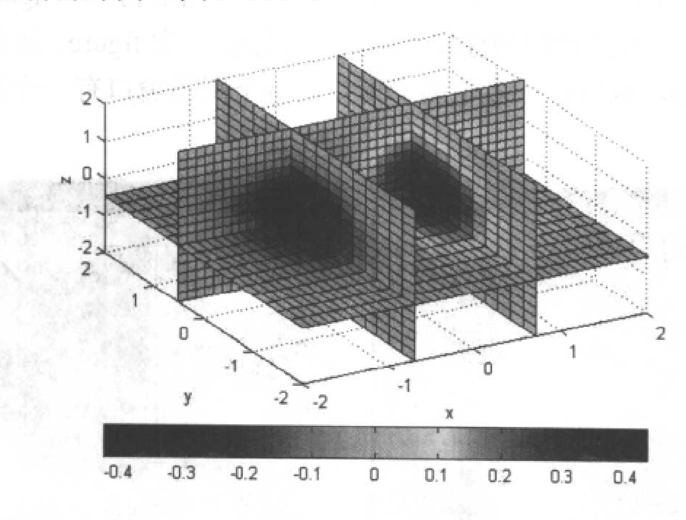


图 5-44　函数的四维表现图

```
>> x=-2:0.1:2; y=-2:0.25:2; z=-2:0.25:2;
>> [X,Y,Z]=meshgrid(x,y,z);
>> V=X.*exp(-X.^2-Y.^2-Z.^2);
>> xi=[-0.7,0.7];
>> yi=0.5;
```

```
>> zi=-0.5;
>> slice(X,Y,Z,V,xi,yi,zi)
>> xlabel('x'); ylabel('y'); zlabel('z');
>> colorbar('horiz')
>> view([-30,45])
```

函数 colorbar 用于产生一个与窗口中表面图形颜色相对应的色彩条，以反映当前坐标轴中图形颜色的刻度，调用 colorbar('horiz')时显示水平色彩条，调用 colorbar('vert')时显示垂直色彩条。

5.4　图形的控制与表现

MATLAB 提供了多种用于图形控制的函数和命令，以便能够更方便清晰地绘制图形。这些函数包括：

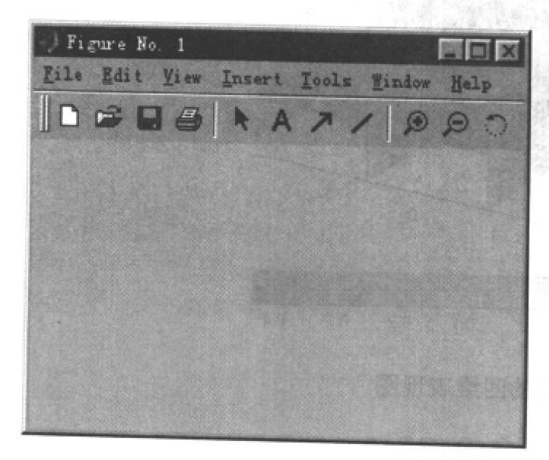
- axis　　手动选择坐标轴范围和比例
- clf　　清图形窗口
- ginput　　利用鼠标的十字基准线输入
- hold　　保持图形
- shg　　显示图形窗口
- subplot　　在图形窗口中绘制子图形

5.4.1　图形窗口

1. 图形窗口的创建和选择

MATLAB 的所有图形都显示在特定的窗口中，称为图形窗口（figure）。

figure 函数用于为当前的绘图创建图形窗口，每运行一次 figure，就会创建一个新的图形窗口。根据绘图的需要，可以打开多个图形窗口，每个图形窗口有一个编号，显示在图形窗口的左上角（图 5-45）。



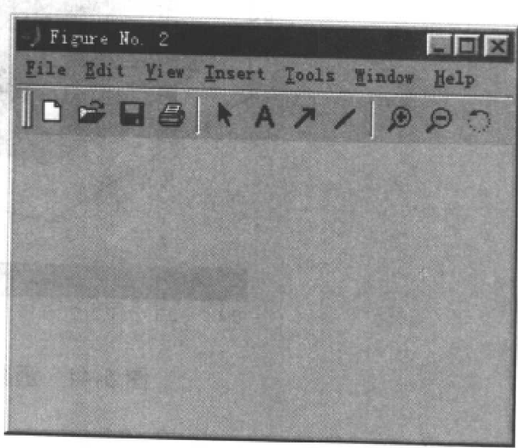


图 5-45　多个图形窗口的自动编号

当调用绘图函数时，如果没有已经打开的图形窗口，则在调用命令的时候自动创建一个新的窗口；如果已有图形窗口，则在缺省情况下直接利用该窗口绘图；如果存在多个图形窗

口，则需要指定将哪一个图形窗口作为当前窗口，如果不指定，MATLAB 将以最后使用过的（或用鼠标最后点中的）窗口作为当前窗口。

figure(*n*)表示将第 *n* 个图形窗口作为当前图形窗口，如果窗口定义了句柄，也可以 figure(*h*) 将句柄为 *h* 的窗口作为当前窗口（指定当前窗口的一个更简单的方法是用鼠标点击需要指定的窗口）。

clc 命令用于清除当前图形窗口中的内容，以便于重新绘图时不发生混淆。

shg 命令用于显示当前图形窗口（将当前图形窗口放在最前面）。

2. 在图形窗口中绘制子图形

在缺省情况下，一个图形窗口中只有一个坐标轴，使用 subplot 函数可以在一个图形窗口上绘制多个图形，其调用格式为：

- subplot(*m*, *n*, *p*)　将图形窗口分成 *m*×*n* 个子窗口并把第 *p* 个子窗口作为当前窗口。子窗口的排列顺序为从左上角开始按行排列。

【例 5-45】　在同一个图形窗口中绘制 4 个子图形。

生成数据点，绘制 4 个子图形（图 5-46）：

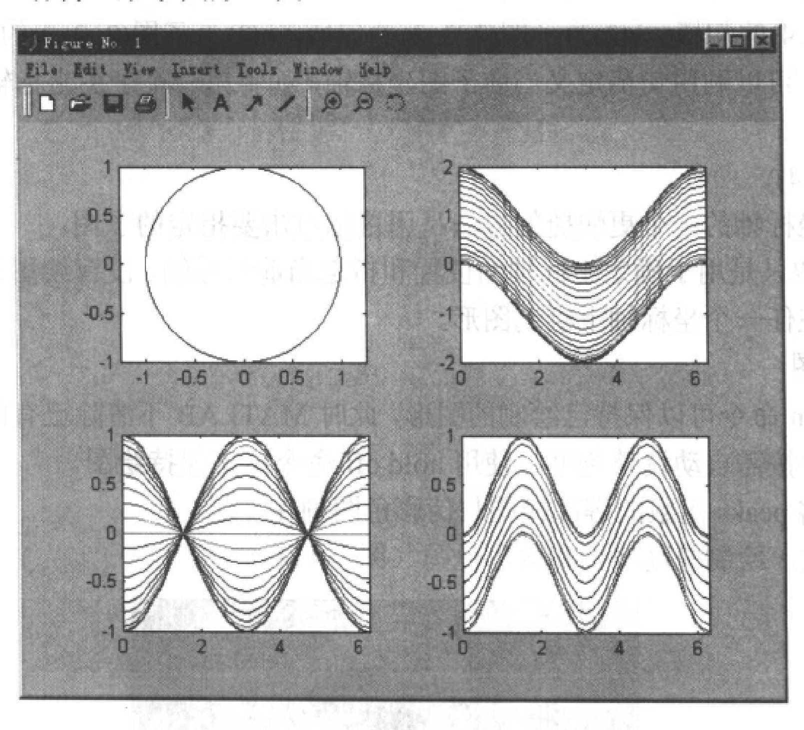


图 5-46　在一个窗口内绘制四个子图形

```
>> t=0:pi/20:2*pi;
>> [x,y]=meshgrid(t);
>> subplot(2,2,1)                              %第 1 个子窗口
>>   plot(sin(t),cos(t)),
>>   axis equal
>> subplot(2,2,2)                              %第 2 个子窗口
>>   z=sin(x)+cos(y);
>>   plot(t,z),
>>   axis([0 2*pi -2 2])
```

```
>> subplot(2,2,3)                          %第3个子窗口
>>    z=sin(x).*cos(y);
>>    plot(t,z),
>>    axis([0 2*pi -1 1])
>> subplot(2,2,4)                          %第4个子窗口
>>    z=sin(x).^2-cos(y).^2;
>>    plot(t,z),
>>    axis([0 2*pi -1 1])
```

在一个图形窗口中绘制多个子图形，实质上是在一个窗口内建立多个坐标轴。当完成绘图后，如果希望对其中的某个子图形做修改或添加内容时，需先将要修改的子图形定义为当前坐标轴。指定当前坐标轴的命令仍然是 subplot(*m*, *n*, *p*)，由 *p* 决定当前坐标轴的序号，也可以用下面的语句获得当前图形窗口中所有子图坐标轴的句柄（有关句柄的内容参见 6.1）：

- *h* = get(*gcf*, 'Children')

h 中包含了所有子图坐标轴的句柄，其中第一个句柄是最后创建的子图坐标轴的句柄，最后一个句柄是第一个创建的子图坐标轴的句柄。对应上面的例子，*h* 中包含 4 个句柄值，*h*(1)为子图(2, 2, 4)的句柄，*h*(2)为子图(2, 2, 3)的句柄，*h*(3)为子图(2, 2, 2)的句柄，*h*(4)为子图(2, 2, 1)的句柄。可以利用句柄定义当前子图坐标轴，如将第一个子图坐标轴定义为当前坐标轴：

- subplot(*h*(4))

指定子图坐标轴的一个更快捷的方法是用鼠标点击要指定的子图。

subplot 函数只是用于确定坐标轴的位置和指定当前坐标轴，没有绘图的功能，可以调用任何绘图函数在任一个坐标轴上绘制图形。

3. 保持绘图

使用 hold on 命令可以保持已绘制的图形，此时 MATLAB 不清除已有的图形，而继续绘制新的图形，并按需自动调整刻度。使用 hold off 命令结束保持绘图。

【例 5-46】　将 peaks 函数的等高线图与伪彩色图画在一起。

生成数据点，绘制等高线图和伪彩色图（图 5-47）：

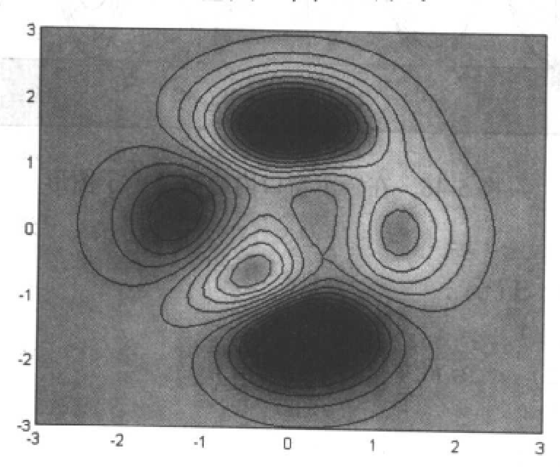


图 5-47　带有等高线的伪彩色图

```
>> [x,y,z]=peaks;                 %产生双变量函数数组
```

```
>> contour(x,y,z,20,'k')        %绘制等高线
>> hold on                      %保持绘图
>> pcolor(x,y,z)                %绘制伪彩色图
>> shading interp               %表面色彩渲染
>> hold off
```

5.4.2 坐标轴控制命令

在缺省情况下，MATLAB 图形坐标轴的状态为：轴的比例控制为“自动(auto)”、坐标轴为“显示(on)”、采用的坐标比例为“直角(xy)”。在这种状态下，用户不需要对图形坐标进行任何干预，坐标刻度范围将根据绘图函数中矢量和矩阵元素值的范围自动确定，这种缺省状态给绘图带来很大方便。

图形是多种多样的，统一的坐标模式不可能总是最有效地表现出所绘图形的特征。因此，MATLAB 设计了控制坐标状态的 axis 函数，axis 函数的调用格式为：

- axis([xmin xmax ymin ymax]) 指定二维图形 x 轴和 y 轴的刻度范围
- axis auto 设置坐标轴为自动刻度（缺省值）
- axis manual（或 axis(axis)） 保持刻度范围不随数据的大小而变化
- axis tight 以数据的大小为坐标轴的范围
- axis ij 设置坐标轴系统的原点在左上角，i 坐标为纵坐标，自上而下为数据增大，j 坐标为横坐标，由左至右为数据增大
- axis xy 使坐标轴回到直角坐标系（缺省值）
- axis equal 使各坐标轴刻度增量相同
- axis square 使各坐标轴长度相同（正方形或立方体），但刻度增量未必相同
- axis normal 自动调节轴与数据的外表比例，使其他设置失效
- axis off 使坐标轴消隐
- axis on 绘制坐标轴（缺省值）

1. 坐标轴的范围

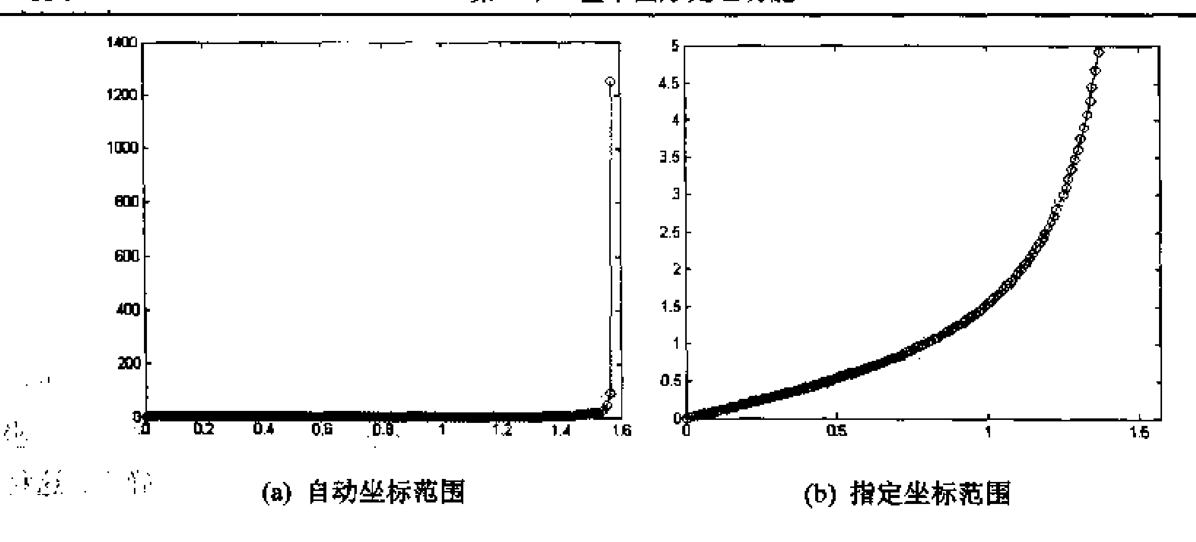
二维图形的坐标轴范围在缺省状态下是根据数据的大小自动设置的，如果需要改变坐标范围，可以通过调用函数 axis([*xmin*,*xmax*,*ymin*,*ymax*])来实现，其中由四个参数组成的矢量分别表示 x 轴的最小值和最大值及 y 轴的最小值和最大值。

【例 5-47】 对比坐标轴范围对正切函数图形表现的影响。

生成数据点，分别按缺省坐标轴范围和指定坐标轴范围绘制曲线（图 5-48）。

```
>> x = 0:.01:pi/2;
>> plot(x,tan(x),'-ro')                  %自动坐标轴范围
>> axis([0,pi/2,0,5])                    %指定坐标轴范围
```

用 axis 函数设定坐标轴范围时要注意，任何一个坐标轴的最小值都必须小于其最大值。



(a) 自动坐标范围　　　(b) 指定坐标范围

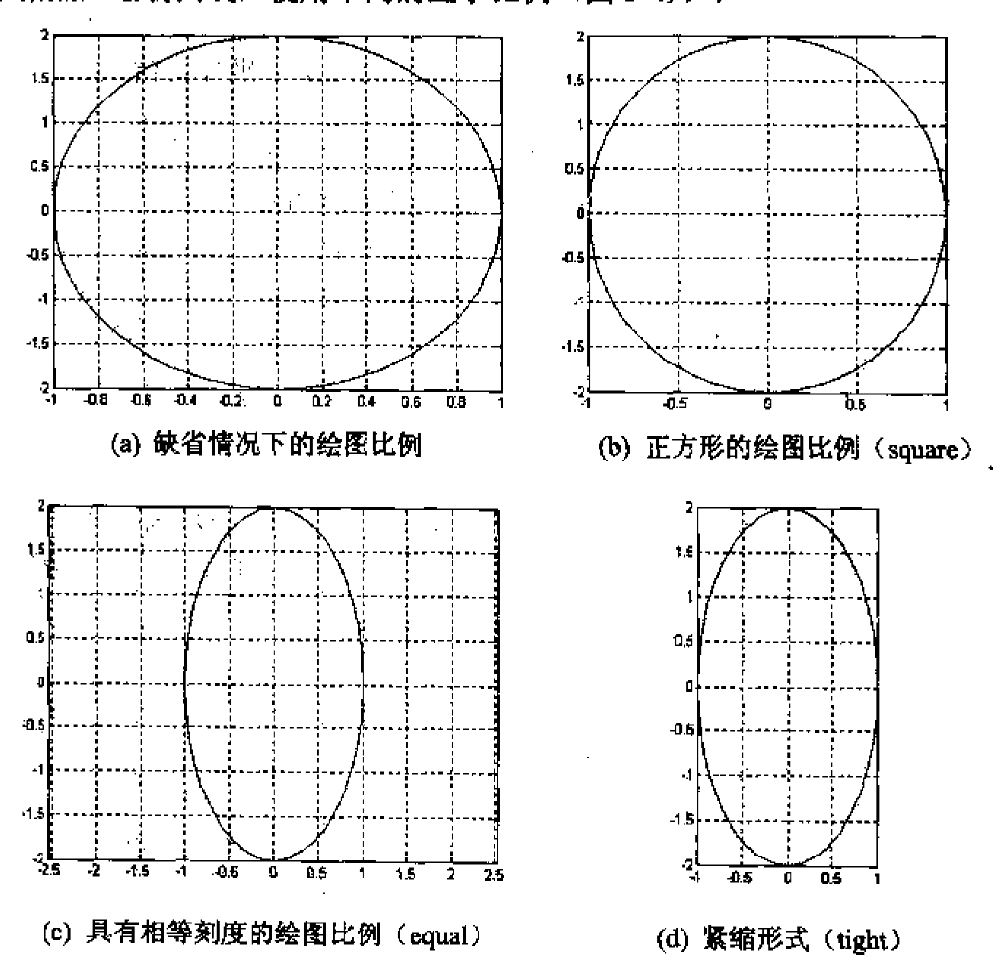
图 5-48　坐标轴范围对图形表现的影响

2. 显示比例对绘图结果的影响

为了能够充分地利用屏幕的显示空间，MATLAB 在缺省情况下按照与屏幕相同的纵横比绘制图形，但为了满足不同情况的需要，MATLAB 还提供了几种不同的显示方式。

【例 5-48】　几种不同的显示方式的比较。

生成数据点，绘制曲线，使用不同的显示比例（图 5-49）：



(a) 缺省情况下的绘图比例　　　(b) 正方形的绘图比例（square）

(c) 具有相等刻度的绘图比例（equal）　　　(d) 紧缩形式（tight）

图 5-49　几种不同的显示方式

```
>> t=0:pi/20:2*pi;
>> plot(sin(t),2*cos(t))                % 缺省值
>> grid on
>> axis square                          % 正方形绘图
>> axis equal                           % X、Y刻度相等
>> axis tight                           % 紧缩形式
```

(1) 本例绘制了一个椭圆，其长轴（y 轴）与短轴（x 轴）之比为 2:1。

(2) 图(a)为缺省状态，没有设置任何坐标轴参数，坐标轴自动按照数据的大小确定刻度，而图形的形状则尽可能按照窗口的比例。这种方式的优点是能够得到最大的图形显示。

(3) 图(b)调用函数 axis square，将两个坐标轴定义为相等的长度（但刻度的增量不一定相等），即绘制正方形坐标下的图形，这种显示方式通常在两个轴具有同样比例（如画圆）时使用。

(4) 图(c)调用函数 axis equal，使得两个坐标轴具有相同的刻度，因此绘制的图形是真实的比例情况。

(5) 图(d)调用函数 axis tight，得到了最紧凑的显示形式，即将坐标轴收缩到图形的尺寸，不留空余空间。

5.4.3 图形的标注

一个好的图形必须有适当的标注，MATLAB 中提供了一系列方便的图形标注函数，这些函数包括：

- title　　图形标题
- xlabel　　x 轴标识
- ylabel　　y 轴标识
- zlabel　　z 轴标识
- text　　任意位置加注文本
- gtext　　鼠标定位加注文本
- legend　　标注图例

图形标注使用的文字可以是字母和数字，MATLAB 6.1 可以使用汉字，也可以按照规定的方法表示希腊字母、数学符号和变形体，例如\pi 表示π，\leq 表示≤，\it 表示斜体字等。

MATLAB 中提供了多种在图形中加入标注的方法，如使用图形窗口中的 Insert 菜单、使用属性编辑器、使用函数输入的方法等，本节只介绍函数输入方法，其他方法参见 6.4.2。

1. 加注坐标轴标识和图形标题

xlabel、ylabel 和 zlabel 函数用于在当前坐标轴上加注 x 轴、y 轴和 z 轴的标识，title 函数用于在当前图形窗口上加标题。这 4 个函数最简单的调用形式是给定一个字符串格式的输入参数，它们的位置自动设定，如 xlabel('时间(t)')表示在 x 轴上标注标识“时间(t)”，title('电压信号')表示将“电压信号”标注为图形的标题。除了最简单的调用方式外，这些函数还可以通过增加属性参数的方法对标识内容做更多的设置。

【例 5-49】　在图形中加注坐标轴标识和标题。

生成数据点，绘制曲线，在图形中加注坐标轴标识和标题（图 5-50）。

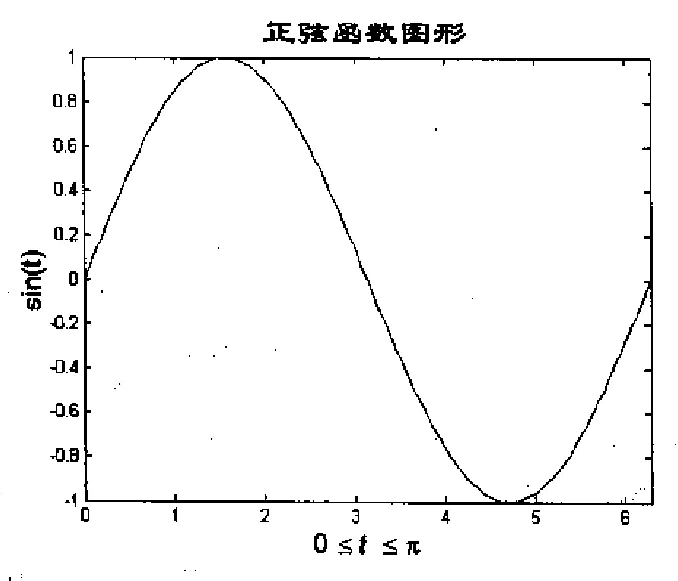


图 5-50　加注轴标识和标题

```
>> t=0:pi/100:2*pi;
>> y=sin(t);
>> plot(t,y)
>> axis([0 2*pi -1 1])
>> xlabel('0 \leq \itt \rm \leq \pi','FontSize',16)
>> ylabel('sin(t)','FontSize',20)
>> title('正弦函数图形','FontName','隶书' ,'FontSize',20)
```

在 xlabel 语句中，\pi 表示希腊字母π，\leq 表示≤，\it 表示后面的字为斜体字，\rm 表示后面的字恢复为正体字。FontSize 表示字体的大小，'FontName'表示字体的类型。

2. 在图形中加注文本

text 函数用于在图形窗口的任何位置加注文本，该函数最简单的调用形式为：

• text(*x*, *y*, '字符串')　*x*、*y* 指定加注文本的位置，text 函数也可以通过指定属性来增加文本形式的变化。

【例 5-50】　在图形中的任意位置加入文本。

在例 5-49 绘制图形的指定位置加入文本（图 5-51）：

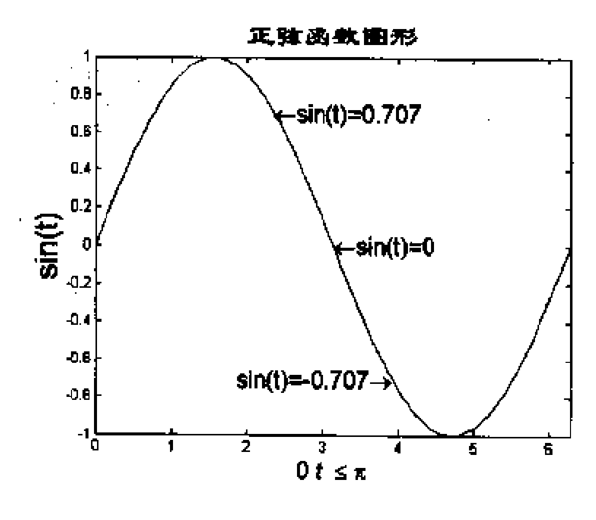


图 5-51　在图形中加注文本

```
>> text(pi,sin(pi),'\leftarrowsin(t)=0','FontSize',16)
>> text(3*pi/4,sin(3*pi/4),'\leftarrowsin(t)=0.707','FontSize',16)
>> text(5*pi/4,sin(5*pi/4),' sin(t)=-0.707\rightarrow',...
       'FontSize',16,'HorizontalAlignment','right')
```

语句中的\leftarrow 表示加一个左箭头，\rightarrow 表示加一个右箭头，HorizontalAlignment 表示水平右对齐。

MATLAB 还提供了一个函数 gtext，该函数用于使用鼠标在指定的位置加注文本，其调用格式为：

- gtext('字符串')

当调用这个函数时，在图形窗口中出现一个随鼠标移动的大十字交叉线，移动鼠标将十字线的交叉点移动到适当的位置，点击鼠标左键，gtext 参数中的字符串就标注在该位置上。

3. 指定 TeX 字符

MATLAB 中的文本对象支持 TeX 字符。使用 TeX 字符可以在文本中使用各种特殊的字母和符号。

【例 5-51】 在图形中使用指定的 TeX 字符。

生成数据，绘制图形，在图形标注中使用指定的 TeX 字符（图 5-52）。

```
>> t=0:pi/100:2*pi;
>> alpha=-0.8;  beta=15;
>> y=sin(beta*t).*exp(alpha*t);
>> plot(t,y)
>> title('{\itAe}^{-\alpha\itt}sin\beta{\itt} \alpha<<\beta')
>> xlabel('时间 \musec.'), ylabel('幅值')
```

title 函数的字符串为：$Ae^{-\alpha t}sin(\beta t)$ $\alpha << \beta$。

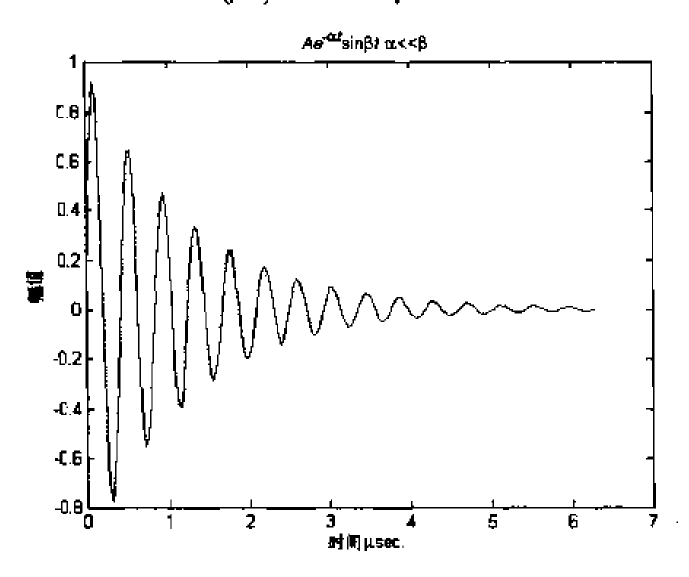


图 5-52 在图形标注中使用指定的 TeX 字符

5.4.4 在图形中添加图例框

使用 legend 函数可以在当前绘图窗口中添加图例，其调用格式为：

• legend(字符串 1,字符串 2，…)　以字符串 1，字符串 2，…作为图形标注的图例。

legend 函数除了创建图例框以外，还可以执行一些预定的功能，如：

• legend('on')　　在当前图形中添加图例框。
• legend('off')　　从当前图形中移走图例框。
• legend('boxon')　　显示图例的边框并使之不透明。
• legend('boxoff')　　不显示图例的边框并使之透明。
• legend('hide')　　使图例框不可见。
• legend('show')　　使图例框可见。
• legend(…，*pos*)　指定图例框显示的位置，图例框被预定了 6 个显示位置：0—取最佳位置，1—右上角（缺省值），2—左上角，3—左下角，4—右下角，-1—图的右侧。

【例 5-52】　在图形中添加图例。

生成数据，绘制图形，在图形中添加图例（图 5-53）。

```
>> x=0:pi/10:2*pi;
>> y1=sin(x);
>> y2=0.6*sin(x);
>> y3=0.3*sin(x);
>> plot(x,y1,x,y2,'-o',x,y3,'-*')
>> legend('曲线 1','曲线 2','曲线 3')
```

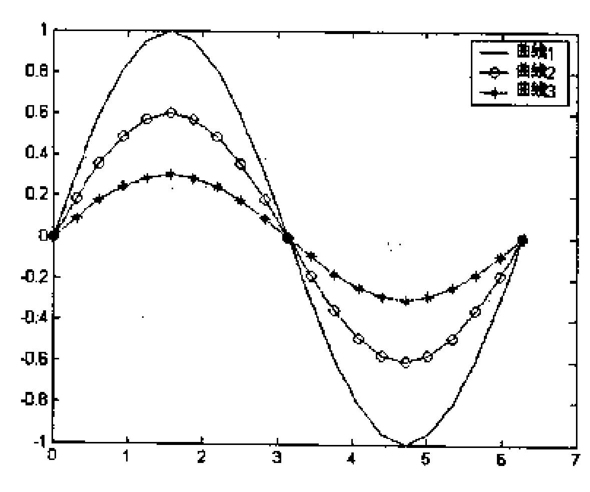


图 5-53　在当前图形中添加图例

习　题　5

5.1　绘制曲线 $y = x^3 + x + 1$，x 的取值范围为[-5,5]。

5.2　有一组测量数据满足 $y = e^{-at}$，t 的变化范围为 0~10。用不同的线型和标记点画出 a=0.1、a=0.2 和 a=0.5 三种情况下的曲线。

5.3　在 5.2 题结果图中添加标题 $y = e^{-at}$，并用箭头线标识出各曲线 a 的取值。

5.4　在 5.2 题结果图中添加标题 $y = e^{-at}$ 和图例框。

5.5　表中列出了 4 个观测点的 6 次测量数据，将数据绘制成为分组形式和叠加形式的条形图。

	第 1 次	第 2 次	第 3 次	第 4 次	第 5 次	第 6 次
观测点 1	3	6	7	4	2	8
观测点 2	6	7	3	2	4	7
观测点 3	9	7	2	5	8	4
观测点 4	6	4	3	2	7	4

5.6 x=[66 49 71 56 38]，绘制饼图，并将第五个切块分离出来。

5.7 $z = xe^{-x^2-y^2}$，当 x 和 y 的取值范围均为-2 到 2 时，用建立子窗口的方法在同一个图形窗口中绘制出三维线图、网线图、表面图和带渲染效果的表面图。

5.8 绘制 peaks 函数的表面图，用 colormap 函数改变预置的色彩图，观察色彩的分布情况。

5.9 用 sphere 函数产生球表面坐标，绘制不透明网格图、透明网格图、表面图和带剪孔的表面图。

5.10 将 5.9 题中的带剪孔的球形表面图的坐标比例变为正方形，以使球面看起来是圆的而不是椭圆的，然后关闭坐标轴的显示。

第 6 章　高级图形处理功能

6.1　图形对象与句柄

6.1.1　图形对象概述

MATLAB 提供了强大的图形处理功能，第 5 章所介绍的基本图形处理功能大多数只使用了 MATLAB 的高级函数，如 plot、mesh、text 等。高级函数一般是对整个图形窗口进行操作的，实际上 MATLAB 还提供了许多用于创建以及操作图形对象的底层函数，底层函数使用户有可能对图形的一个或几个对象进行独立操作，而不影响图形的其他部分，正是这种功能为图形处理提供了极大的灵活性。

MATLAB 的图形对象是指图形系统中最基本、最底层的单元，这些对象包括根屏幕（Root）、图形窗口（Figures）、坐标轴（Axes）、线（Lines）、块（Patches）、面（Surfaces）、图像（Images）、文本（Text）、控件单元（Uicontrols）和菜单单元（Uimenus）等。根据各对象的相互关系可以构成如图 6-1 所示的树状层次。

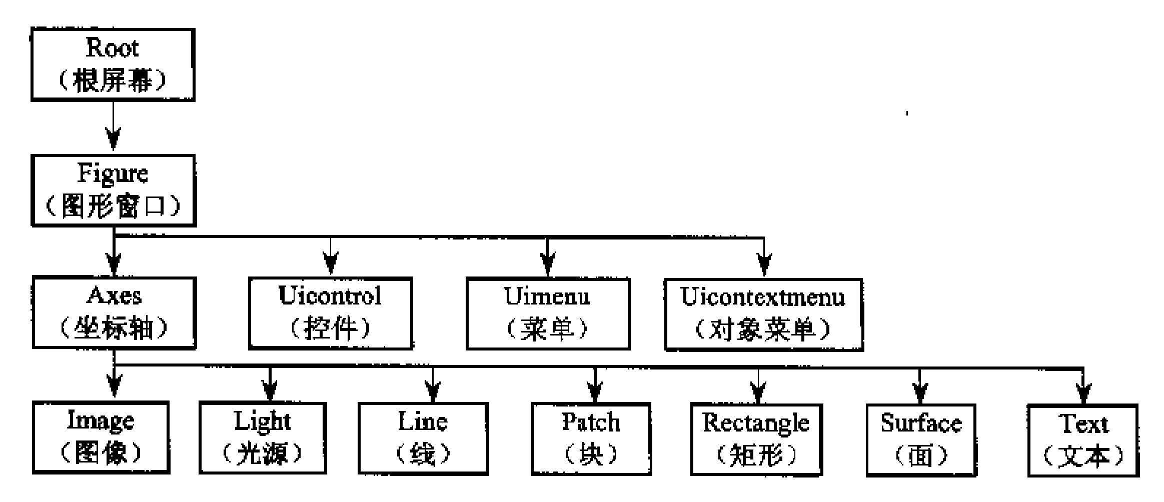


图 6-1　图形对象的层次

各层次图形对象是相互关联的，它们之间的关系为父子或兄弟关系，一般要先创建父对象，然后再建立子对象和孙对象。例如要绘制一个线对象，则需要建立一个轴对象，而建立轴对象，就需要先创建一个图形窗口。

6.1.2　图形对象的类型

1. 根（Root）对象

根对象位于图形对象的最顶层，相当于计算机屏幕。根对象只有一个，而所有其他图形对象都是它的子孙，根对象不能被建立，启动 MATLAB 时它就自动创建了，但是可以设置

根对象的属性以控制图形的显示。

2. 图形窗口（Figure）对象

图形窗口对象是独立于根对象的用于显示图形的窗口，图形窗口对象是根对象的子对象，而所有其他图形对象都是图形窗口的子孙。

在调用所有的绘图函数（如 plot 和 surf）时，如果不存在图形窗口对象，MATLAB 会自动建立一个图形窗口，如果在根对象下有多个图形窗口，总有一个图形窗口被指定为“当前图形窗口”，并作为建立与修改图形的目标对象。

3. 坐标轴（Axis）

坐标轴对象在图形窗口中定义一个区域，并确定该区域中子对象的方向或角度。坐标轴对象是图形窗口的子对象，同时又是图像、灯光、线、块、表面和文字等图形对象的父对象。

如果不存在坐标轴对象，所有的绘图函数（如 plot、surf、mesh 和 bar 等）都会建立坐标轴对象。如果在图形窗口中有多个坐标轴对象，总有一个是当前坐标轴，并且是其子对象的目标（uicontrol 和 uimenu 不是坐标轴对象的子对象）。

4. 控件（Uicontrol）

控件对象用于图形用户界面的设计，当驱动对象时执行回调程序。控件对象有多种类型，如按钮、列表框、滚动条等。

5. 菜单（Uimenu）

菜单对象是图形窗口的下拉菜单。当用户选择一个独立的菜单项时执行回调程序，用户定义的 uimenu 被放置在图形窗口的菜单条上所有已存在的菜单项的右侧。

6. 对象菜单（Uicontextmenu）

当用鼠标右键点击某一个图形对象时弹出对象菜单，对象菜单必须与某一个图形对象联合起来使用。

7. 图像（Image）

MATLAB 的图像对象由数据矩阵和色图矩阵组成。有 3 种基本的图像类型：索引图、强度图和真彩色图。由于图像是二维的，因此只能以缺省的二维视角观察图像。

8. 光源（Light）

光源对象定义光源，它影响所有位于轴上的块对象和面对象。光源对象是看不到的，可以通过属性设置控制光源的颜色、位置和其他公共属性。

9. 线（Line）

线对象是大多数二维和三维绘图的基础，高级绘图函数 plot、plot3 和 loglog 等用于建立线对象，其父对象坐标轴对象的坐标系统决定了线的位置和方向。

10. 块（Patch）

块对象是带有边界且内部被填充的多边形，一个单个的块可以包含多个表面，每一个表面独立地使用固定或插值的色彩着色。fill、fill3 和 contour3 建立块对象，父对象坐标轴对象的坐标系统决定了块对象的位置和方向。

11. 矩形（Rectangle）

矩形对象是二维填充区域，矩形对象的形状在矩形和椭圆之间变化。对于建立流程图类型的绘图，矩形对象是非常有用的。

12. 面（Surface）

面对象用于矩阵数据的三维表现，每个矩阵元素的值作为 x-y 平面上的高度进行绘图。面对象是由矩阵数据所规定顶点位置的四边形的组合，MATLAB 可以用固定或插值的色彩绘制面对象，也可以只用网格线连接顶点。父对象坐标轴对象的坐标系统决定了表面的位置和方向。高级函数 pcolor、surf 和 mesh 等用于建立面对象。

13、文本（Text）

文本对象是字符串，父对象坐标轴对象的坐标系统决定了文本的位置。高级函数 title、xlabel、ylabel、zlabel 和 gtext 用于建立文本对象。

6.1.3　图形句柄

使用 MATLAB 的图形对象句柄（Handle）可以对各种图形对象进行更为细腻的修饰和控制，不仅可以绘制复杂的图形，而且为动态图形的绘制奠定了基础。

在 MATLAB 的图形系统中，所有的图形操作都是针对图形对象而言的，每个对象从创建时起就被赋予了唯一的标识，这种标识就是该对象的句柄。

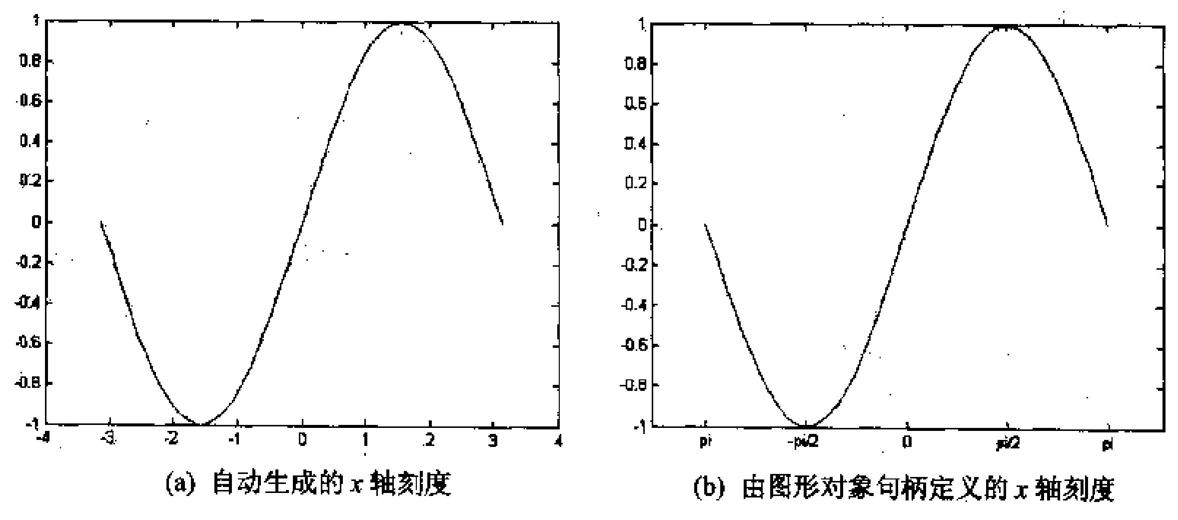
图形对象的句柄是 MATLAB 进行图形处理和建立图形用户界面的基础。句柄的值可以是一个数，也可以是一个矢量，例如每个计算机根对象只有一个，它的句柄总是 0，图形窗口的句柄总是正整数，它标识了图形窗口的序号，而等高线函数（contour）创建的句柄则是一组值，每个值代表等高线中的一条，也就是说每一条线就有一个句柄值。

利用句柄既可以操纵一个已经存在的图形对象的属性，也可以在建立图形对象时指定属性的值，特别是对指定图形对象句柄的操作不会影响同时存在的其他图形对象，这一点是非常重要的。

set 和 get 函数用于设置和获取指定句柄的图形对象的属性值，如 set(*h*,'属性名','属性值')设置句柄 *h* 的图形对象的属性，具体属性由属性名和属性值决定；get（*h*,'属性名','属性值'）获得句柄为 *h* 的图形对象指定属性的当前值，如果没有设置句柄，可以使用 gcf 命令获得当前图形窗口的句柄，使用 gca 命令获得当前坐标轴对象句柄，使用 gco 命令获得当前对象的句柄。下面通过一些简单的例子了解图形对象句柄的特点及其用法。

【例 6-1】　自定义坐标轴的显示刻度。

首先，使用 plot 函数绘制曲线，然后通过图形对象句柄定义坐标轴的显示刻度（图 6-2）：



(a) 自动生成的 x 轴刻度　　　　(b) 由图形对象句柄定义的 x 轴刻度

图 6-2　坐标刻度对比

```
>> t=-pi:pi/20:pi;
>> y=sin(t);
>> plot(t,y)
>> set(gca,'xtick',[-pi:pi/2:pi],…
     'xticklabel',{'-pi','-pi/2','0','pi/2','pi'})
```

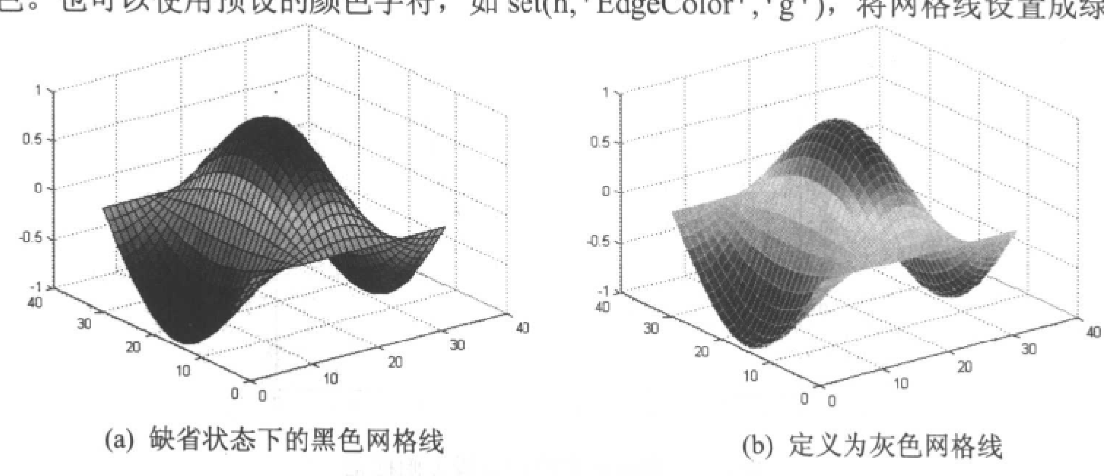
本例中使用 gca 获得当前坐标轴对象的句柄，然后用'xtick'属性设置 x 轴刻度的位置（从-pi~pi，间隔 pi/2，共设置 5 个点），用'xticklabel'来指定刻度的值，由于通常习惯于用角度（或弧度）度量三角函数，因此本例中使用{'-pi', '-pi/2', '0', 'pi/2', 'pi'}表示 5 个刻度值。

【例 6-2】 改变三维网格图中网格线的颜色。

首先建立句柄为 h 的三维曲面，然后通过设置图形对象句柄的参数改变网格线的颜色（图 6-3）：

```
>> h=surf(sphere(30));
>> set(h,'EdgeColor',[0.8,0.8,0.8])
```

缺省状态下曲面的网格线为黑色，通过设置图形对象句柄的参数将曲面的网格线设置成浅灰色。也可以使用预设的颜色字符，如 set(h,'EdgeColor','g')，将网格线设置成绿色。

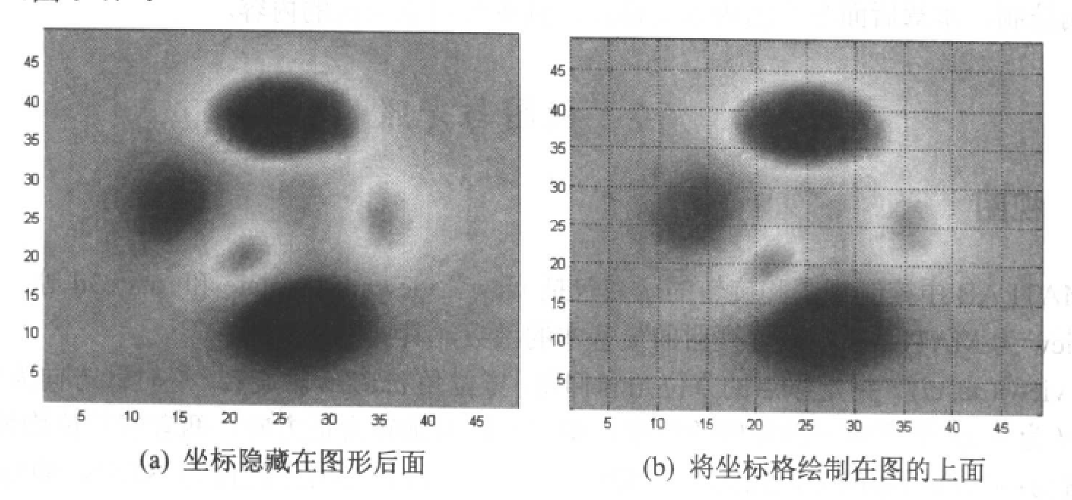


(a) 缺省状态下的黑色网格线　　(b) 定义为灰色网格线

图 6-3 改变网格线的颜色

【例 6-3】 在图形的上面绘制坐标格。

首先，绘制三维曲面的伪彩色图，然后通过设置图形对象句柄的参数在图形的上面绘制坐标格（图 6-4）：



(a) 坐标隐藏在图形后面　　(b) 将坐标格绘制在图的上面

图 6-4 在图形的上面绘制坐标格

```
>> pcolor(peaks);
>> shading interp
>> grid on
>> set(gca,'Layer','top')
```

【例 6-4】　用鼠标确定曲线上点的位置。

首先，绘制二维曲线，然后通过设置图形对象句柄的鼠标响应获得点的坐标，根据所获得点的 x 坐标在曲线上绘制星号（图 6-5）：

```
>> t=0:pi/20:2*pi; plot(t,cos(t)); hold on
>> set(gcf,'WindowButtonDownFcn',...
    'p=get(gca,''CurrentPoint'');',...
    'px=p(1,1);py=cos(px);'...
    'plot(t,cos(t));'...
    'plot(px,py,''*r'');'])
>> axis(axis)
```

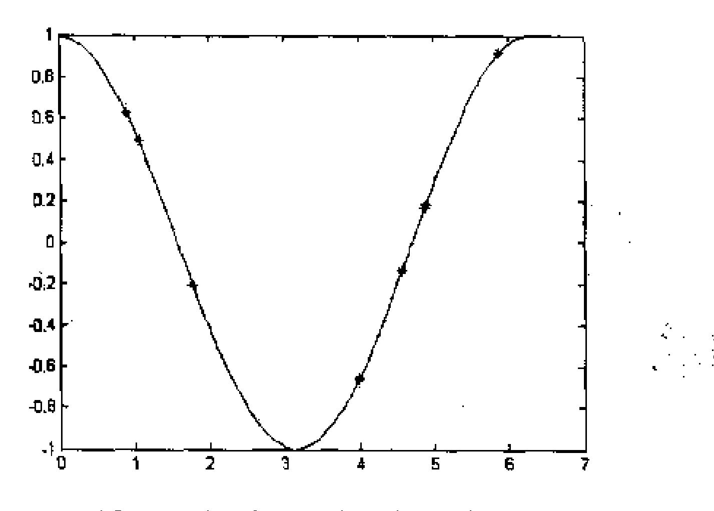


图 6-5　由鼠标定位在曲线上画标记

这段程序用于响应鼠标动作，当用户在图形窗口内移动并点击鼠标时，对应于鼠标点所确定的 x 轴的坐标自动在曲线上绘制一个星号。

使用图形对象句柄的地方很多，特别是在图形用户界面的设计中，所有的控件对象都要由句柄控制，本章后面各节的内容也都涉及到图形对象句柄的内容。

6.2　视图与光照

6.2.1　视图

MATLAB 中控制视图和旋转的函数包括 view、viewmtx、rotate 和 rotate3d 等。

view 为 MATLAB 中设置视图的最基本的函数，其调用格式为：

• view (*az, el*)　指定原点的方位角和仰角，方位角 *az* 绕 *z* 轴旋转，逆时针方向为正方向，仰角 *el* 为从 *xy* 平面向 *z* 轴旋转的角度，朝向 *z* 轴的旋转为正方向，两者的单位均为度。缺省设置为：二维图形的方位角为 0°，仰角为 90°；三维图形的方位角为-37.5°，仰角为 30°。

- view ([*vx*, *vy*, *vz*])　返回视点的直角坐标。
- view (2)　使用缺省的二维视图（方位角为 0°，仰角为 90°）。
- view (3)　使用缺省的三维视图（方位角为–37.5°，仰角为 30°）。

【例 6-5】　绘制不同视图的视觉效果的图形。

绘制双峰函数并变换各种视图（图 6-6）：

```
>> surf(peaks(50))
>> view(90,0)
>> view(30,45)
>> view(2)
```

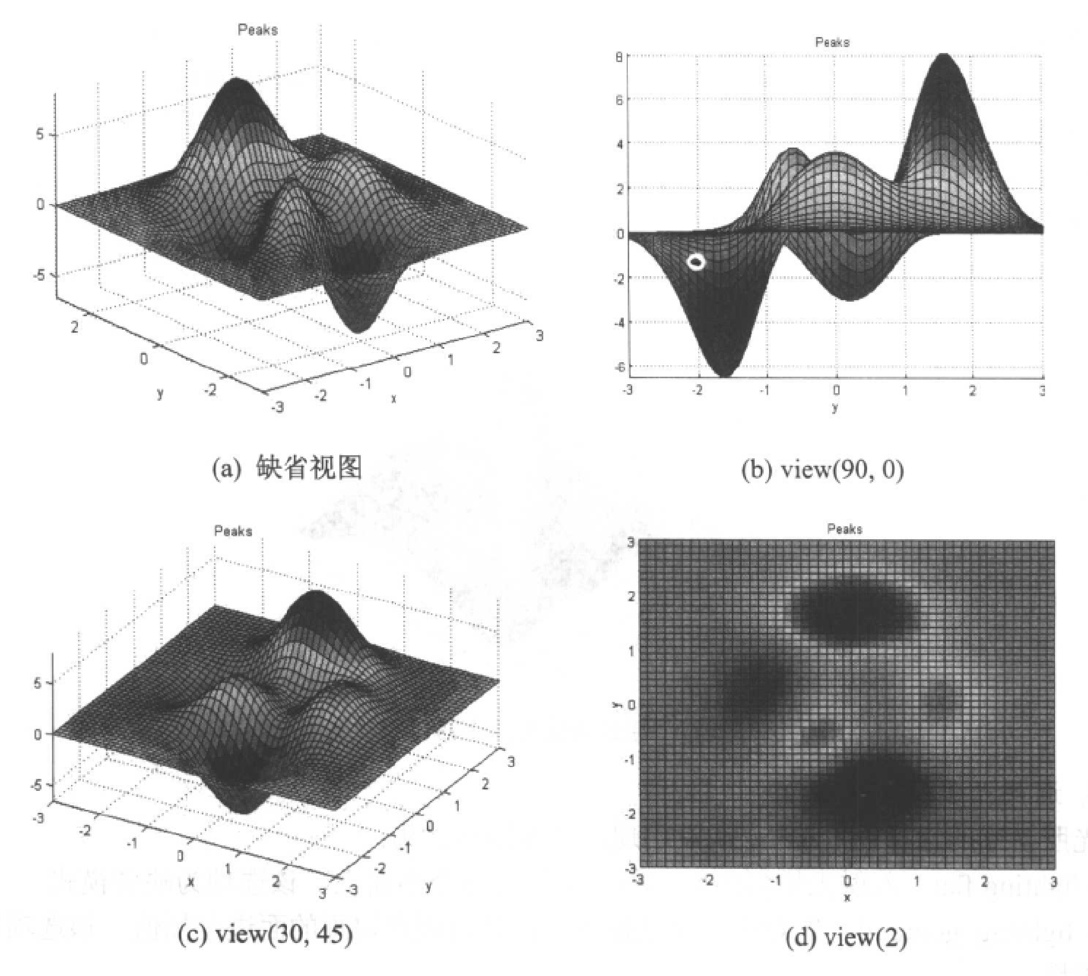


图 6-6　不同视图视觉效果的对比

6.2.2　光照

MATLAB 提供了许多对图形的光照进行控制的函数，通过使用这些函数，可以使绘制的图形具有更好的视觉效果。

1. 创建光源对象

light 函数用于创建光源对象，其调用格式为：

- light('*PropertyName*', *PropertyValue*, ...)　在当前坐标轴对象中创建光源对象，'*PropertyName*' 主要包括 Color、Position 和 Style 等属性，Color 的值用 *ColorSpec* 确定，缺

省值为白色；Position 由[*x y z*]坐标确定，缺省值为[1 0 1]；Style 的值为 infinite（无限远处的平行光源）或 local（点光源），缺省值为 infinite。

- *handle* = light(...)　返回所创建光源的图形对象句柄。

【例 6-6】　通过创建光源增强显示效果。

绘制表面图，通过创建光源增强显示效果（图 6-7）：

```
>> [x,y]=meshgrid(-8:.1:8);
>> r=sqrt(x.^2+y.^2)+eps;
>> z=2*sin(r)./r;
>> surf(x,y,z)
>> shading interp;
>> daspect([5 5 1])                    % 调整显示的纵横比
>> axis tight
>> view(-50,30)
>> light('position',[-10 -10 2])
```

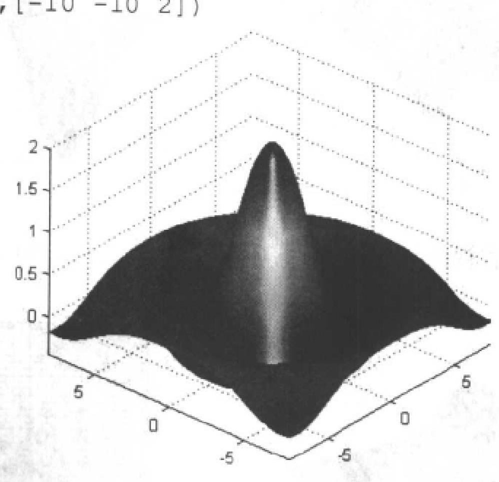


图 6-7　通过创建光源增强显示效果

2. 选择光照模式

光照模式的选择由 lighting 函数实现，其调用格式为：

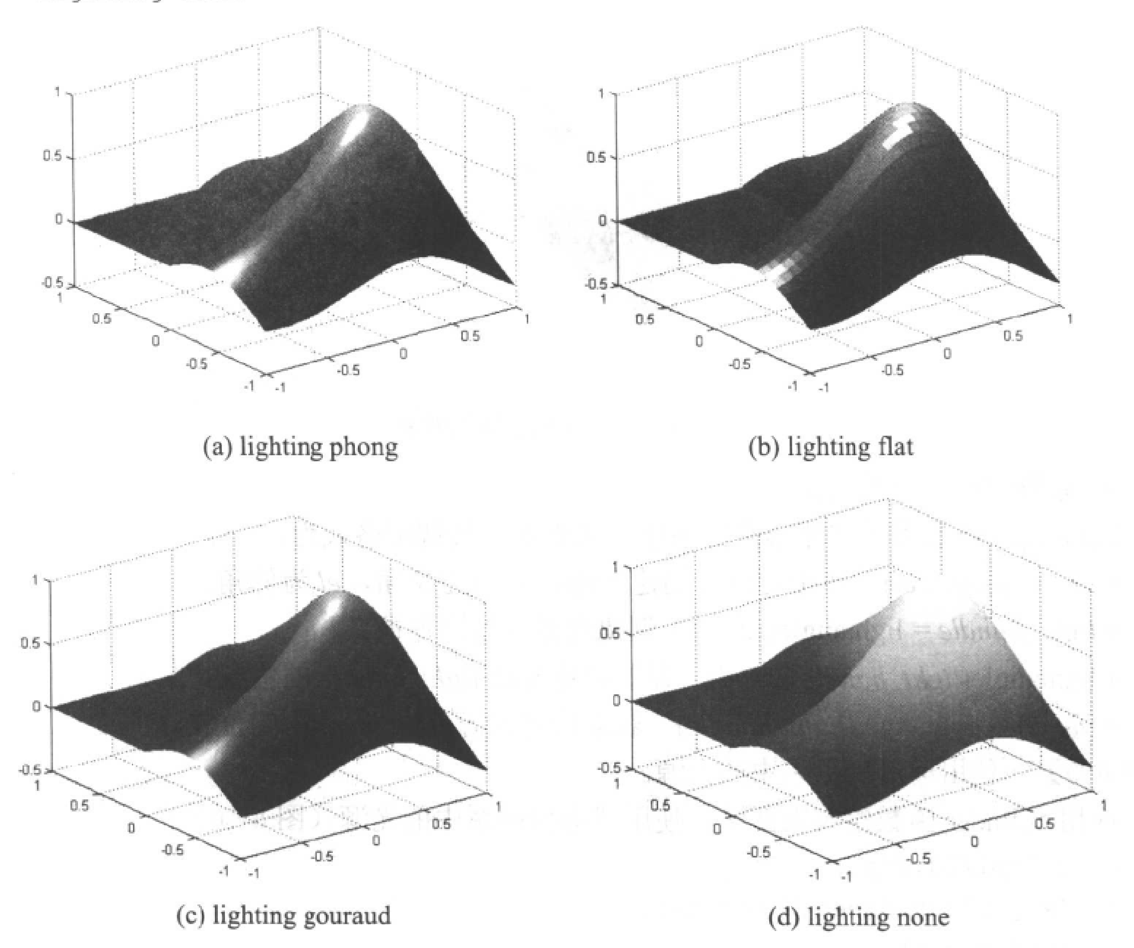
- lighting flat　入射光均匀照射在图形对象的每个小面上，该选项为缺省模式。
- lighting gouraud　先对顶点颜色插值，再对由顶点勾画的面进行插值。该选项用于曲面的表现。
- lighting phong　对顶点处法线插值，再计算各像素点的反光。该选项表现效果最好，但占用机时较多。
- lighting none　关闭光照。

【例 6-7】　不同光照模式的效果对比。

应用 membrane 函数绘制表面图，使用不同的光照模式进行对比（图 6-8）：

```
>> membrane
>> shading flat
>> light
>> lighting phong
>> lighting flat
```

```
>> lighting gouraud
>> lighting none
```



(a) lighting phong (b) lighting flat

(c) lighting gouraud (d) lighting none

图 6-8 不同光照模式的效果对比

3. 照相机光源

MATLAB 提供了一组照相机坐标下的图形处理函数，其中 camlight 用于创建照相机坐标系统的光源对象，其调用格式为：

- camlight headlight　在照相机位置创建光源。
- camlight right　在照相机的右上方创建光源。
- camlight left　在照相机的左上方创建光源。
- camlight　缺省情况，在照相机的右上方创建光源。
- camlight(*az*, *el*)　在相对照相机的方位角为 *az*、仰角为 *el* 的位置创建光源。
- camlight(...,'style')　设置光源的类型，infinite 或 local。
- camlight(*light_handle*, ...)　设置指定光源 *light_handle* 的位置。
- *light_handle* = camlight(...)　返回光源对象的图形对象句柄。

【例 6-8】 展示照相机光源的效果。

应用 sphere 函数绘制表面图，使用照相机光源的效果（图 6-9）：

```
>> surf(sphere(40))
>> shading flat
>> camlight('infinite')
```

```
>> lighting phong
```

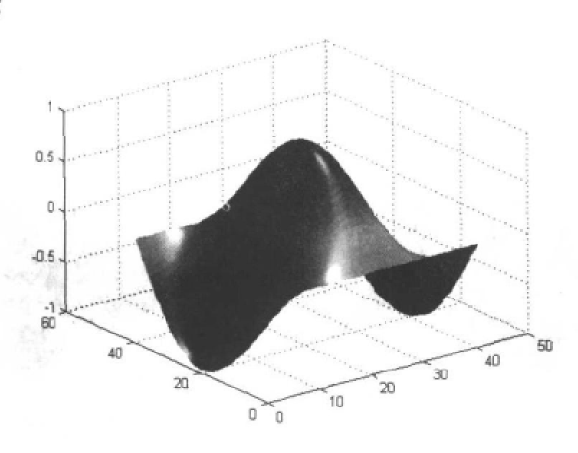


图 6-9　照相机光源的效果

4. 球形坐标系中的光源

lightangle 函数用于在球形坐标系中创建光源，其调用格式为：

- lightangle(*az*, *el*)　在指定位置创建光源，*az* 为方位角，*el* 为仰角。
- *light_handle* = lightangle(*az*, *el*)　创建光源并返回句柄。
- lightangle(*light_handle*, *az*, *el*)　设置指定光源 *light_handle* 的位置。
- [*ax el*] = lightangle(*light_handle*)　获取指定光源 *light_handle* 的位置。

【例 6-9】　使用球形坐标系中的光源。

应用 cylinder 函数绘制表面图，使用球形坐标系中的光源（图 6-10）：

```
>> t=0:pi/20:2*pi;
>> [x y z]=cylinder(2+sin(t));
>> surf(x,y,z)
>> shading flat
>> lightangle(-50,30)
```

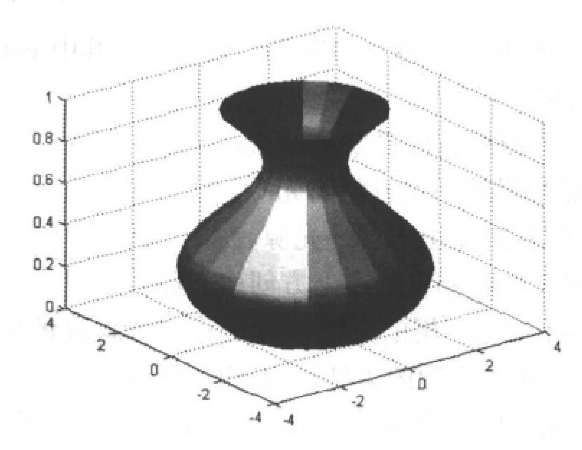


图 6-10　使用球形坐标系中的光源

5. 材料反射系数

使用 material 函数可以设置材料的反射系数，以获得更好的光照效果，其调用格式为：

- material shiny　有光泽、明亮，镜反射份额较大。
- material dull　暗淡、柔和，漫反射份额较大，无镜面亮点。

- material metal　有金属光泽，镜反射份额大，背景光和漫反射份额小。
- material([*ka kd ks n sc*])　对 5 大反射要素进行特别设置。5 大反射要素分别为：

  ka　环境光（背景光）强度（数值越大，光越强）；

  kd　漫反射强度（数值越大，光越强）；

  ks　镜面反射光强度（数值越大，光越强）；

  n　镜面指数（控制镜面亮点大小，指数越大，亮点越小）；

  sc　镜面颜色的反射系数。
- material default　缺省设置。

【例 6-10】　镜面反射和漫反射效果的对比。

应用 sphere 函数绘制球，使用不同的镜面反射和漫反射强度（图 6-11）。

```
% 镜面反射强度为 0,1,2（沿 x 轴方向变化）
% 漫反射强度为 0,0.5,1（沿 y 轴方向变化）
[X,Y,Z]=sphere(40);
ks=[0,1,2];
kd=[0,0.2,0.4];
for i=1:3
   for j=1:3
      subplot('position',[0.33*(j-1),0.33*(i-1),.33,.33])
      surf(X,Y,Z);
      shading interp
      colormap([0.9 0.9 0.9])
      light ('position',[-3,-2,5],'style','local')
      lighting phong
      material([0.5,kd(i),ks(j),20,0.3])
      axis square off
   end
end
```

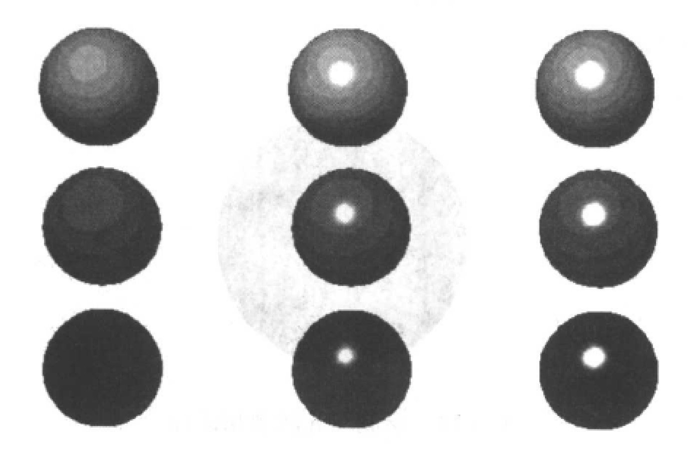


图 6-11　镜面反射和漫反射效果的对比。

【例 6-11】　光照和材料反射系数。

应用 cylinder 函数绘制柱面，设置光源和材料反射系数（图 6-12）"

```
>> clf
```

```
>> t=linspace(0,2*pi,100);
>> r=1-exp(-t/2).*cos(4*t);                %旋转母线
>> [x,y,z]=cylinder(r,60);                 %产生旋转柱面数据
>> ii=find(x<0&y<0);                       %确定 x-y 平面第四象限上的数据下标
>> z(ii)=NaN;                              %剪切
>> surf(x,y,z)
>> colormap(pink)
>> shading interp
>> light('position',[-3,-1,3],'style','local')       %设置光源
>> material([0.5,0.4,0.3,10,0.3])                    %设置表面反射
```

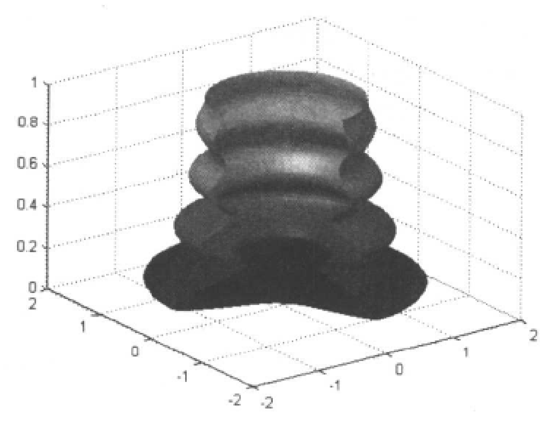


图 6-12　光照和材料反射系数

【例 6-12】　绘制双光源效果的球。

应用 sphere 函数绘制球，设置两个光源和材料反射系数（图 6-13）。

```
>> sphere(36);
>> axis equal
>> light('Position',[1 3 2]);
>> light('Position',[-3 -1 3]);
>> material shiny
>> axis off
```



图 6-13　绘制双光源效果的球

6.3　体积可视化

MATLAB 提供了一组体积可视化的函数，利用这组函数可以实现复杂的三维内剖图、

流线流管图的绘制。

6.3.1 流线图

streamline 函数使用二维或三维数据绘制流线图，其调用格式为：

- h = streamline(X, Y, Z, U, V, W, *startx*, *starty*, *startz*) 绘制三维矢量数据 U、V、W 的流线图并返回图形对象的句柄，数组 X、Y、Z 定义了 U、V、W 的坐标，*startx*、*starty*、*startz* 定义了流线图的起点位置。
- h = streamline(U, V, W, *startx*, *starty*, *startz*) 绘制三维矢量数据 U、V、W 的流线图并返回图形对象的句柄，假定数组 X、Y、Z 由[X,Y,Z]= meshgrid(1:N, 1:M, 1:P) 定义，其中 [M, N, P] = size(U)。
- h = streamline(X, Y, Z) X、Y、Z 为单元数组，该数组是由顶点数组计算得到的。
- h = streamline(X, Y, U, V, *startx*, *starty*) 绘制二维矢量数据 U、V 的流线图，数组 X、Y 定义了 U、V 的坐标，*startx*、*starty* 定义了流线图的起点位置，h 为图形对象句柄。
- h = streamline(U, V, *startx*, *starty*) 绘制二维矢量数据 U、V 的流线图并返回图形对象的句柄。假定数组 X、Y 由[X, Y]= meshgrid(1:N, 1:M) 定义，其中[M, N] = size(U)。
- h = streamline(XY) XY 为单元数组，该数组是由顶点数组计算得到的。
- h = streamline(..., *options*) 指定绘制流线图的选项，*options* 可以定义为只有一个元素的矢量（步长）或有两个元素的矢量（步长和流线顶点的最大个数）：

[*stepsize*] 或 [*stepsize*, *max_number_vertices*]

如果无指定值，MATLAB 将使用缺省值：

stepsize = 0.1，*max_number_vertices* = 1 000

【例 6-13】 利用 MATLAB 中所提供的数据集 *wind* 绘制流线图。

调入数据 *wind*，创建变量并绘制流线图（图 6-14）：

```
>> load wind
>> [sx,sy,sz] = meshgrid(80,20:10:50,0:5:15);
>> h = streamline(x,y,z,u,v,w,sx,sy,sz);
>> set(h,'Color','red')
>> view(3)
```

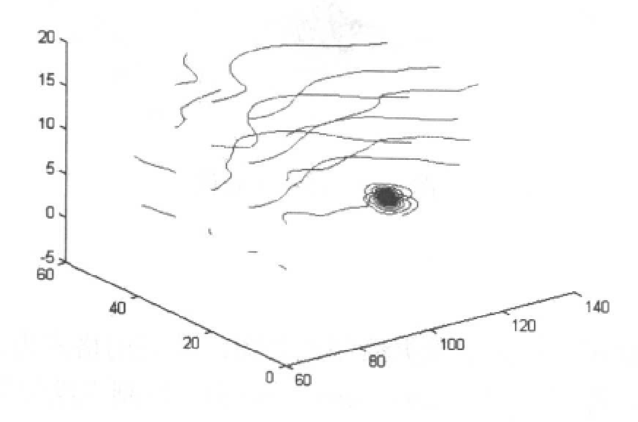


图 6-14 绘制流线图

6.3.2　流点图

streamparticles 函数使用二维或三维数据绘制流点图，其调用格式为：

- streamparticles(*vertices*)　绘制矢量的流点图，流点由标记代表，可以显示流线的位置和速率，*vertices* 为一个二维或三维矢量的单元数组。
- streamparticles(*vertices*, *n*)　用 *n* 确定需要绘制多少流点。
- streamparticles(..., '*PropertyName*', *PropertyValue*,...)　使用指定的属性和指定值控制流点。流点的属性包括：Animate——流点动画，可以设置动画重复的次数；FrameRate——动画频率，即每秒的动画幅数；ParticleAlignment——流点是否与流线对齐。
- streamparticles(*line_handle*, ...)　使用由 *line_handle* 定义的图形对象绘制流点图。
- *h* = streamparticles(...)　返回图形对象句柄。

【例 6-14】　利用 MATLAB 中所提供的数据集 *wind* 绘制流点图。

调入数据 *wind*，绘制流线图和流点动画图（图 6-15）：

```
load wind
[sx sy sz] = meshgrid(80,20:1:55,5);
verts = stream3(x,y,z,u,v,w,sx,sy,sz);
sl = streamline(verts);
iverts = interpstreamspeed(x,y,z,u,v,w,verts,.025);
axis tight; view(30,30); daspect([1 1 .125])
camproj perspective; camva(8)
set(gca,'DrawMode','fast')
box on
streamparticles(iverts,35,'animate',10,'ParticleAlignment','on')
```

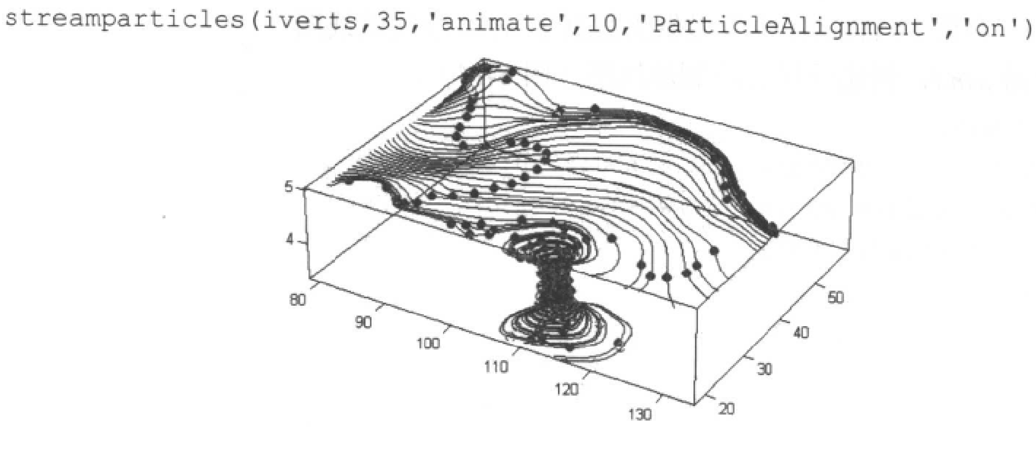


图 6-15　绘制流点图

6.3.3　流带图

streamribbon 函数使用三维矢量数据绘制流带图，其调用格式为：

- streamribbon(*X*, *Y*, *Z*, *U*, *V*, *W*, *startx*, *starty*, *startz*)　绘制三维矢量数据 *U*、*V*、*W* 的流带图，数组 *X*、*Y*、*Z* 定义了 *U*、*V*、*W* 的坐标，*startx*、*starty*、*startz* 定义了流带图的起点位置。
- streamribbon(*U*, *V*, *W*, *startx*, *starty*, *startz*)　绘制三维矢量数据 *U*、*V*、*W* 的流带图，假

定数组 *X*、*Y*、*Z* 由[*X*, *Y*, *Z*]= meshgrid(1:*N*, 1:*M*, 1:*P*) 定义，其中[*M*, *N*, *P*] = size(*U*)。

- streamribbon(*vertices*, *X*, *Y*, *Z*, *cav*, *speed*) 预先计算的流线顶点、扭曲角速度和流速。*vertices* 为流线顶点的单元数组，*X*、*Y*、*Z*、*cav* 和 *speed* 为三维数组。
- streamribbon(*vertices*, *cav*, *speed*) 数组 *X*、*Y*、*Z* 由[*X*, *Y*, *Z*]= meshgrid(1:*N*, 1:*M*, 1:*P*) 定义，其中[*M*, *N*, *P*] = size(*cav*)。
- streamribbon(*vertices*, *twistangle*) 使用矢量的单元数组 *twistangle* 确定流带的扭曲，*vertices* 与 *twistangle* 的每一对应元素的大小必须相同。
- streamribbon(..., *width*) 设置流带的宽度。
- *h* = streamribbon(...) 返回图形对象句柄。

【例 6-15】 利用 MATLAB 中所提供的数据集 *wind* 绘制流带图。

调入数据 *wind*，绘制流带图（图 6-16）：

```
load wind
[sx sy sz] = meshgrid(80,20:10:50,0:5:15);
daspect([1 1 1])
streamribbon(x,y,z,u,v,w,sx,sy,sz);
axis tight
shading interp;
view(3);
camlight; lighting gouraud
```

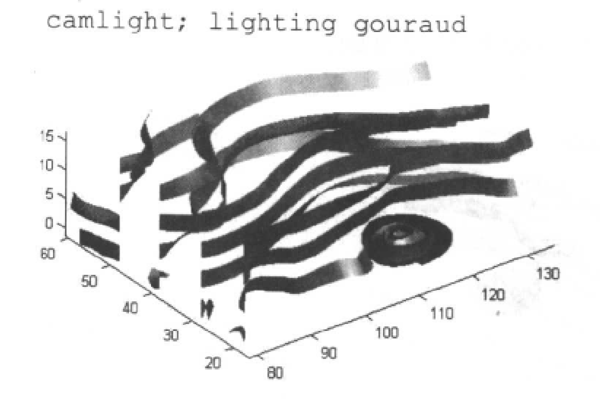


图 6-16 绘制流带图

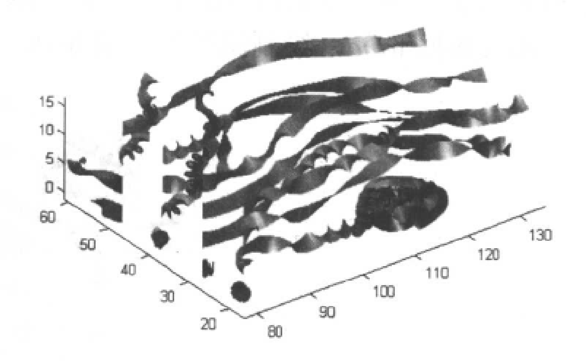


图 6-17 绘制扭曲流带图

【例 6-16】 利用 MATLAB 中所提供的数据集 *wind* 绘制扭曲流带图。

调入数据 *wind*，绘制扭曲流带图（图 6-17）：

```
load wind
[sx sy sz] = meshgrid(80,20:10:50,0:5:15);
daspect([1 1 1])
verts = stream3(x,y,z,u,v,w,sx,sy,sz);
cav = curl(x,y,z,u,v,w);
spd = sqrt(u.^2 + v.^2 + w.^2).*.1;
streamribbon(verts,x,y,z,cav,spd);
axis tight
shading interp
view(3)
camlight; lighting gouraud
```

6.3.4 流管图

streamtube 函数使用三维数据绘制流管图，其调用格式为：

- streamtube(*X*, *Y*, *Z*, *U*, *V*, *W*, *startx*, *starty*, *startz*)　绘制三维矢量数据 *U*、*V*、*W* 的流管图，数组 *X*、*Y*、*Z* 定义了 *U*、*V*、*W* 的坐标，*startx*、*starty*、*startz* 定义了流管图的起点位置。
- streamtube(*U*, *V*, *W*, *startx*, *starty*, *startz*)　绘制三维矢量数据 *U*、*V*、*W* 的流管图，假定数组 *X*、*Y*、*Z* 由[*X*, *Y*, *Z*]= meshgrid(1:*N*, 1:*M*, 1:*P*) 定义，其中[*M*, *N*, *P*] = size(*U*)。
- streamtube(*vertices*, *X*, *Y*, *Z*, *divergence*)　预先计算的流线顶点和差异。*vertices* 为流线顶点的单元数组，*X*、*Y*、*Z*、*cav* 和 *speed* 为三维数组。
- streamtube(*vertices*, *divergence*)　数组 *X*、*Y*、*Z* 由[*X*, *Y*, *Z*]= meshgrid(1:*N*, 1:*M*, 1:*P*) 定义，其中[*M*, *N*, *P*] = size(*divergence*)。
- streamtube(*vertices*, *width*)　使用矢量的单元数组 *width* 确定流带的宽度，*vertices* 与 *width* 的每一对应元素的大小必须相同，*width* 也可以是一个标量，定义所有流管的宽度。
- streamtube(*vertices*)　自动选择宽度。
- streamtube(..., [*scale n*])　按 *scale* 参数确定流管的宽度，*n* 为沿圆管周边分布的点数，缺省时 *scale* = 1，*n* = 20。
- *h* = streamtube(...)　返回图形对象句柄。

【例 6-17】　利用 MATLAB 中所提供的数据集 *wind* 绘制流管图。

调入数据 *wind*，绘制流管图（图 6-18）：

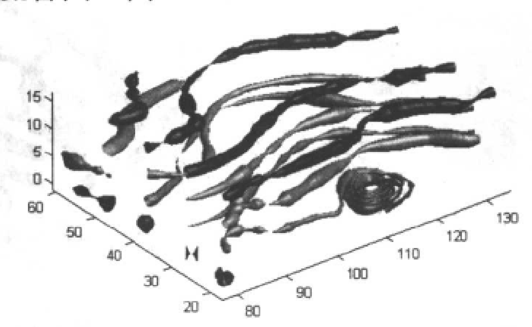


图 6-18　绘制流管图

```
load wind
[sx sy sz] = meshgrid(80,20:10:50,0:5:15);
daspect([1 1 1])
streamtube(x,y,z,u,v,w,sx,sy,sz);
view(3)
axis tight
shading interp;
camlight; lighting gouraud
```

6.4 图形窗口的功能及设置

MATLAB 6 的图形窗口除提供了图形绘制的区域外，还提供了丰富的属性设置能力和很强的图形处理功能。本节将对这些设置和功能做简要介绍。

6.4.1 图形复制参数设置

图形复制参数设置用于控制 MATLAB 的图形复制过程（图 6-19）。

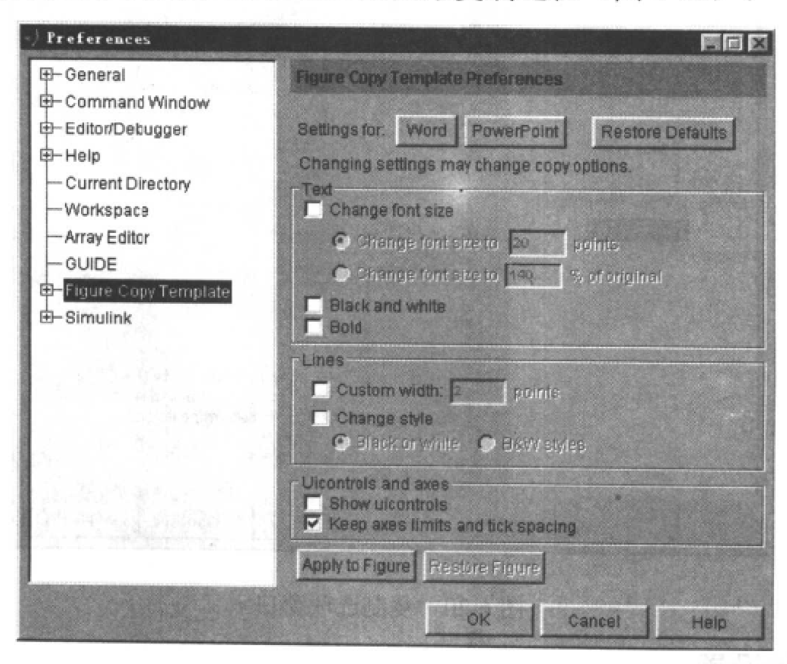


图 6-19 图形复制设置窗口

在该窗口中，第一行选择 MATLAB 的图形输出为 Word 或 Powerpoint 文档时的设置，Restore Dafaults 恢复缺省设置。

其他图形复制的选项如下：

• Text（文本） 包括 3 个选项，分别为“Change font size”（修改字体尺寸）、“Black and white”（黑白）和 Bold（粗体字）。如果勾选了“Change font size”，将激活两个单选框，分别为直接输入字体的大小（单位为 point，1 point=1/72 inch）和输出前后的比例。

• Lines（线） 包括两个选项，分别为“Custom width”（线宽，单位为 point）和“Change style”（改变线型），线型的选择分别为“Black and White”（黑白模式）或“B&W styles”（用不同线型区分不同的曲线）。

• Uicontrols and axes（用户控件和坐标轴） 包括两个选项，分别为“Show uicontrols”（选择是否显示用户控件）和“Keep axes limits and tick spacing”（选择是否保持坐标轴范围和刻度）。

在树状结构中展开“Figure Copy Template”项，可以设置“Copy Options”（复制选项）（图 6-20）。

• Clipboard format（剪贴板格式）　包括 3 个选项，分别为 Metafile（以 EMF 格式复制图形）、“Preserve information”（选择 EMF 格式，并在可能的情况下使用该格式）、Bitmap（以 BMP 格式复制图形）。

• Figure backgroud color（设置复制图形的背景颜色）　包括 3 个选项，分别为“Use figure color”（保留原图形的背景颜色）、“Force white backgroud”（强制使用白色背景）、“Transparent background”（使用透明背景颜色）。

• Size（设置尺寸）　“Match figure screen size”（保持图形原来的大小）。

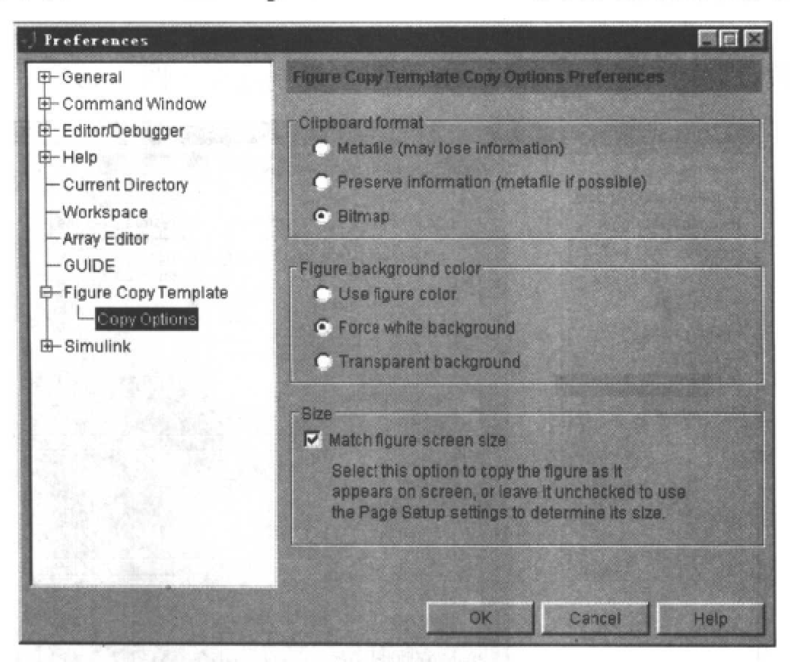


图 6-20　复制选项窗口

6.4.2　图形对象设置

在 5.4 中介绍了图形的控制与表现，完成这些控制都是通过命令和函数来实现的。MATLAB 还提供了另一种更方便地完成图形的控制与表现的方法，即通过图形化的参数设置窗口完成图形对象的设置。根据对象的不同，设置的内容也不同，下面逐一进行介绍。

1. 图形窗口（Figure）对象

在图形窗口的 Edit 菜单中选择“Figure Properties…”，打开图形窗口对象参数设置窗口（图 6-21），使用该窗口可以对图形窗口对象的参数进行设置。

图形窗口对象的参数设置包括 4 个页面，分别为：

• Style（外观）　包括设置图形背景色、是否显示菜单等。

• Title（名称）　图形窗口名称。

• Rendering（渲染）　绘图的表现方式，可以在 painters、zbuffer 和 OpenGL 中选择，还可以选择是否使用备份存储和双缓冲区等。

• Info（信息）　包括设置图形对象的标识、图形窗口是否显示、选择设置父对象或子对象等。

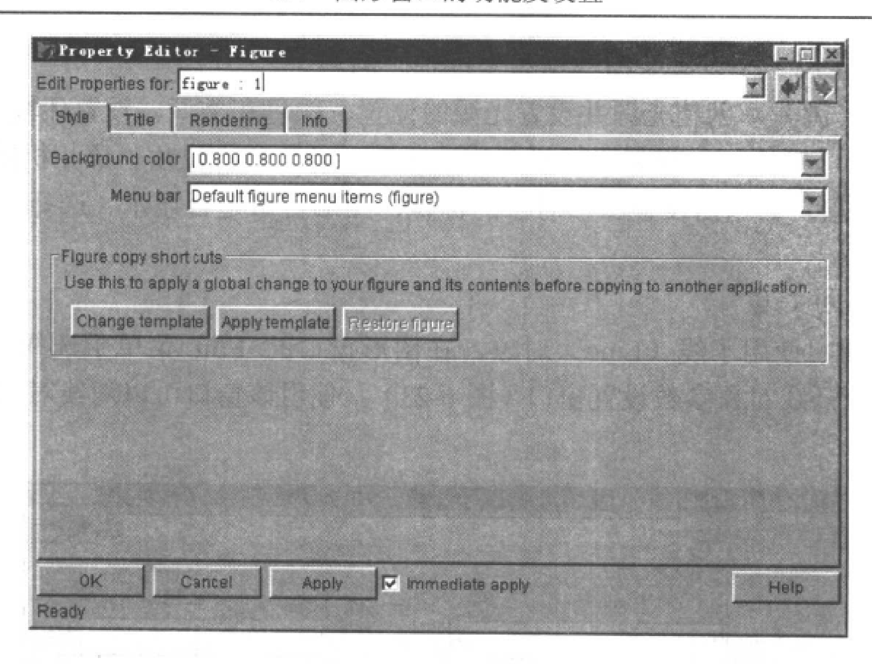


图 6-21 图形窗口对象设置窗口

2. 坐标轴（Axes）对象

在图形窗口的 Edit 菜单中选择“Axes Properties...”，打开坐标轴对象参数设置窗口（图 6-22），使用该窗口可以对坐标轴对象的参数进行设置。

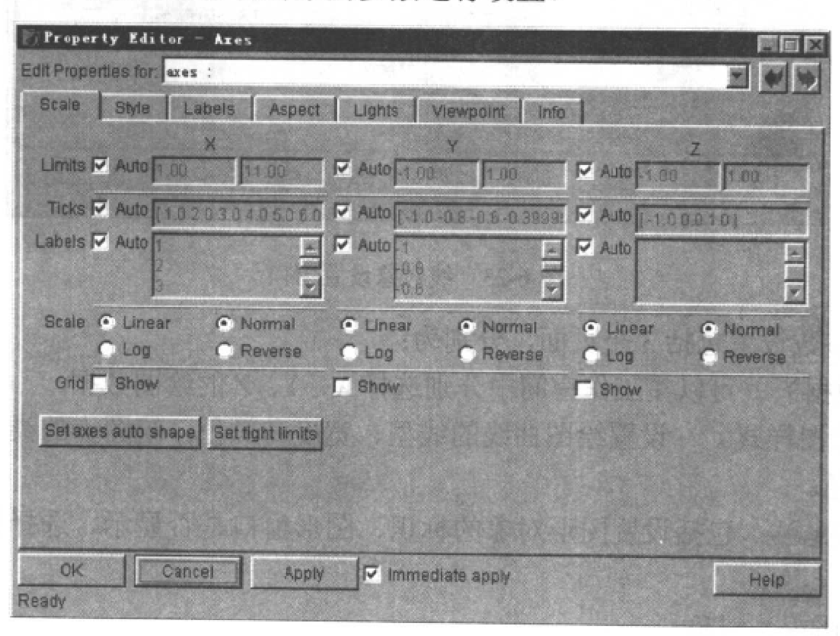


图 6-22 坐标轴对象设置窗口

坐标轴对象的参数设置包括 7 个页面，分别为：

• Scale（刻度） 分别设置 3 个坐标轴的绘图范围、刻度间隔、刻度值、坐标类型（线性或对数）、坐标轴方向、是否绘制坐标网格等。

• Style（外观） 设置是否绘制轴框、绘制坐标轴线的尺寸、绘图区和各坐标轴的颜色、坐标刻度值的字体和尺寸、坐标刻度值的位置（上下或左右）等。

• Labels（标题） 设置图形标题和各坐标轴标识。

- Aspect（比例）　设置绘图数据和坐标轴比例。
- Lights（光源）　创建光源并设置光源的位置、颜色和类型，设置环境光的颜色。
- Viewpoint（视图）　设置正投影图或透视图、仰角和方位角、照相机的参数等。
- Info（信息）　包括设置图形对象的标识、图形窗口是否显示、选择设置父对象或子对象等。

3. 线（Line）对象

如果在绘图中使用了线（Line）对象，在图形窗口的 Edit 菜单中选择 Current Object Properties...，打开线对象参数设置窗口（图 6-23），使用该窗口可以对线对象的参数进行设置。

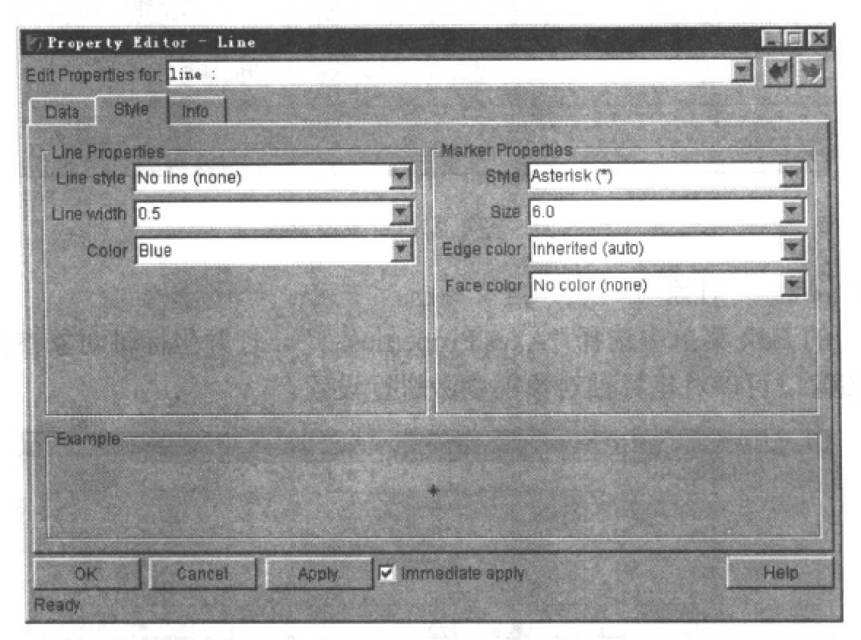


图 6-23　线对象设置窗口

线对象的参数设置包括 3 个页面，分别为：

- Data（数据）　可以在工作空间中分别选择 X、Y、Z 的绘图数据。
- Style（绘图样式）　设置绘图曲线的线型、宽度和颜色，绘图点的类型、尺寸、边缘颜色和填充颜色。
- Info（信息）　包括设置图形对象的标识、图形窗口是否显示、选择设置父对象或子对象等。

4. 面（Surface）对象

如果在绘图中使用了面（Surface）对象，在图形窗口的 Edit 菜单中选择 “Current Object Properties...”，打开面对象参数设置窗口（图 6-24），使用该窗口可以对面对象的参数进行设置。

面对象的参数设置包括 6 个页面，分别为：

- Data（数据）　可以在工作空间中分别选择 X、Y、Z 的绘图数据，同时选择颜色数据。
- Style（绘图样式）　设置绘图曲面的颜色、光照和透明处理，网格线的类型、宽度、颜色、网格方式、光照和透明处理，绘图方式（面、线、点），绘图点的类型、尺寸、边缘颜色和填充颜色等。

- Color（颜色） 设置颜色、色彩图、图线颜色、绘图点颜色等。
- Transparency（透明处理） 设置曲面和网格的透明度。
- Lighting（光照） 设置光照参数、材料反射系数、轴光等。
- Info（信息） 包括设置图形对象的标识、图形窗口是否显示、选择设置父对象或子对象等。

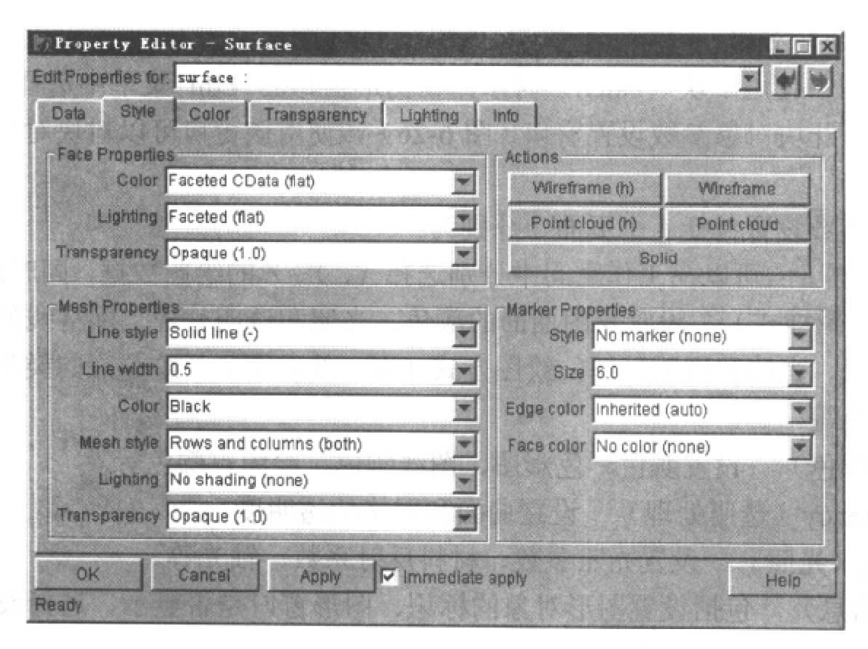


图 6-24 面对象设置窗口

5. 文本（Text）对象

如果在绘图中使用了文本（Text）对象，在图形窗口的 Edit 菜单中选择“Current Object Properties...”，打开文本对象参数设置窗口（图 6-25），使用该窗口可以对文本对象的参数进行设置。

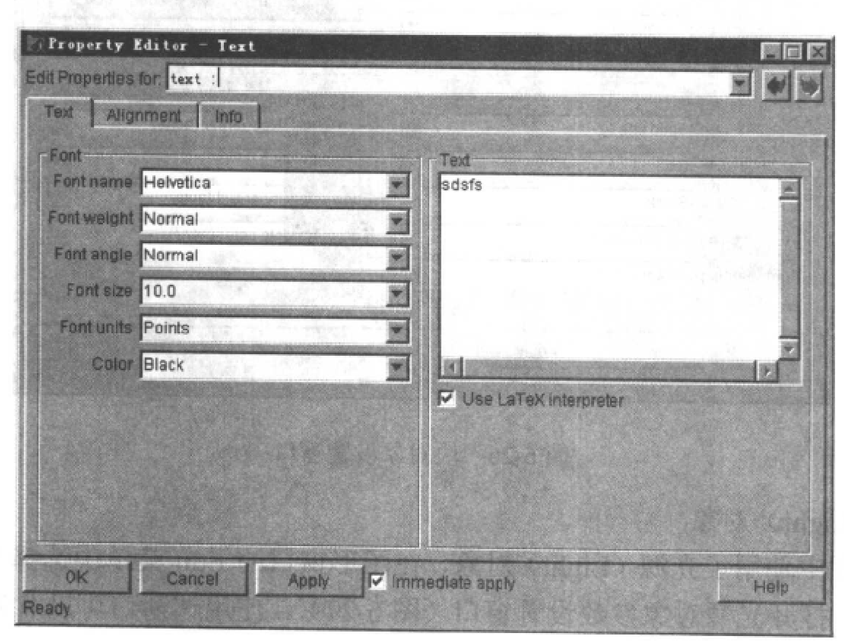


图 6-25 文本对象设置窗口

文本对象的参数设置包括 3 个页面，分别为：

• Text（文本）　设置文本字体的名称、尺寸、颜色和文本的内容。

• Alignment（对齐）　设置文本的位置和对齐方式。

• Info（信息）　包括设置图形对象的标识、图形窗口是否显示、选择设置父对象或子对象等。

6. 块（Patch）对象

如果在绘图中使用了块（Patch）对象，在图形窗口的 Edit 菜单中选择 Current Object Properties...，打开块对象参数设置窗口（图 6-26），使用该窗口可以对块对象的参数进行设置。

块对象的参数设置包括 6 个页面，分别为：

• Data（数据）　可以在工作空间中分别选择 X、Y、Z 的绘图数据，同时选择颜色数据。

• Style（绘图样式）　设置绘图曲面的颜色、光照和透明处理，网格线的类型、宽度、颜色、网格方式、光照和透明处理，绘图方式（面、线、点），绘图点的类型、尺寸、边缘颜色和填充颜色等。

• Color（颜色）　设置颜色、色彩图、图线颜色、绘图点颜色等。

• Transparency（透明处理）　设置曲面和网格的透明度。

• Lighting（光照）　设置光照参数、材料反射系数、轴光等。

• Info（信息）　包括设置图形对象的标识、图形窗口是否显示、选择设置父对象或子对象等。

图 6-26　块对象设置窗口

7. 光源（Light）对象

如果在绘图中使用了光源（Light）对象，在图形窗口的 Edit 菜单中选择“Current Object Properties...”，打开光源对象参数设置窗口（图 6-27），使用该窗口可以对光源对象的参数进行设置。

光源对象的参数设置包括两个页面，分别为：

- Light（光源） 设置光源的位置、颜色和类型。
- Info（信息） 包括设置图形对象的标识、图形窗口是否显示、选择设置父对象或子对象等。

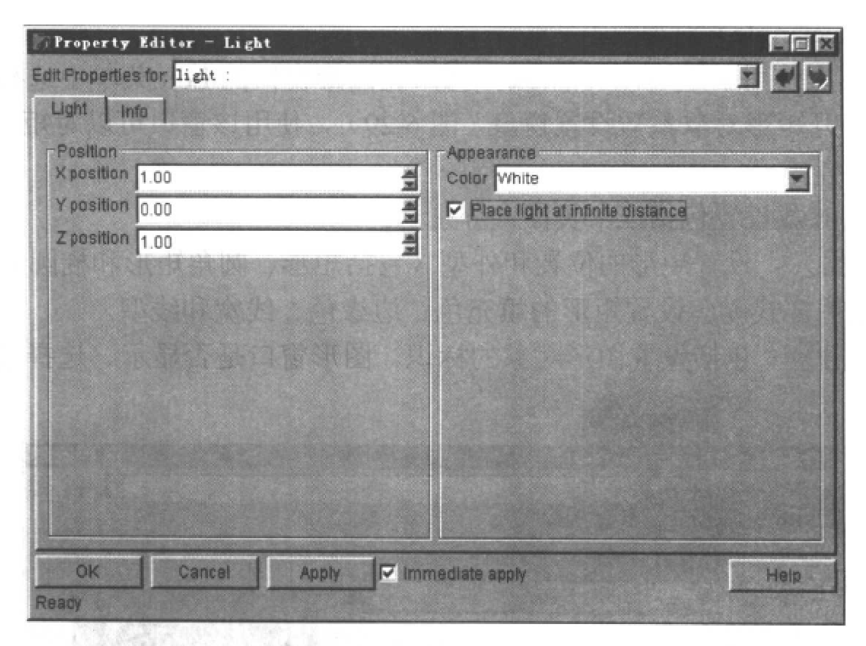


图 6-27 光源对象设置窗口

8. 图像（Image）对象

如果在绘图中使用了图像（Image）对象，在图形窗口的 Edit 菜单中选择 Current Object Properties...，打开图像对象参数设置窗口（图 6-28），使用该窗口可以对图像对象的参数进行设置。

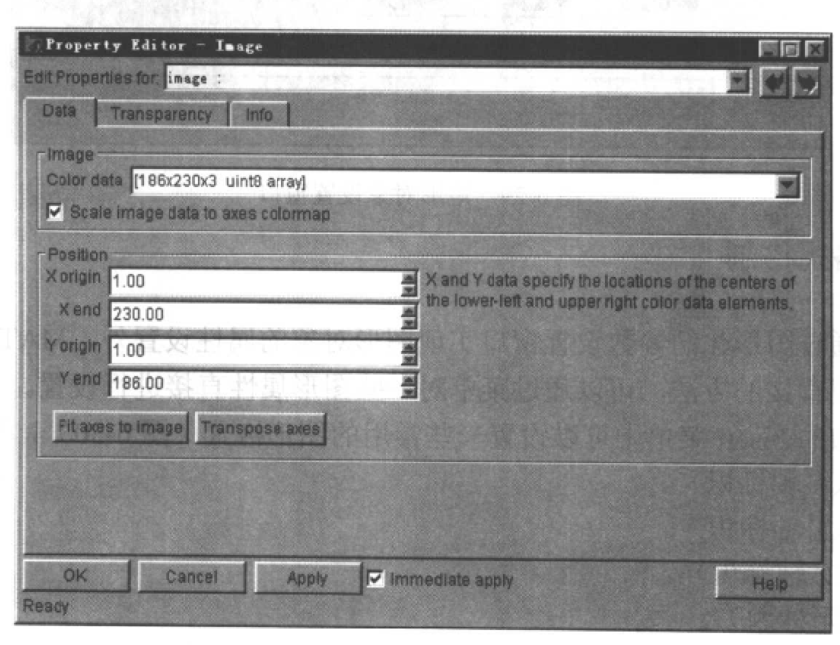


图 6-28 图像对象设置窗口

图像对象的参数设置包括 3 个页面，分别为：

- Data（数据） 选择图像数据、设置图像显示位置。

• Transparency（透明处理）　设置透明处理方法。

• Info（信息）　包括设置图形对象的标识、图形窗口是否显示、选择设置父对象或子对象等。

9. 矩形（Rectangle）对象

如果在绘图中使用了矩形(Rectangle)对象，在图形窗口的 Edit 菜单中选择 Current Object Properties...，打开矩形对象参数设置窗口（图 6-29），使用该窗口可以对矩形对象的参数进行设置。

矩形对象的参数设置包括三个页面，分别为：

• Data（数据）　设置矩形的位置和外型（包括矩形、圆角矩形和椭圆）。

• Style（绘图样式）　设置矩形的填充色、边缘色、线宽和线型。

• Info（信息）　包括设置图形对象的标识、图形窗口是否显示、选择设置父对象或子对象等。

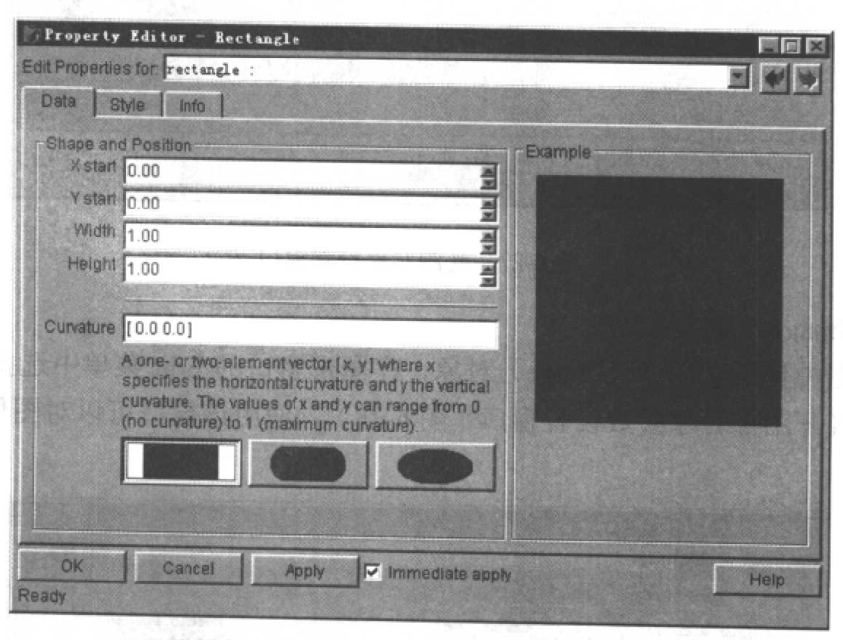


图 6-29　矩形对象设置窗口

6.4.3　图形格式控制

除了可以通过图形化的参数设置窗口实现图形对象的属性设置外，MATLAB 的图形窗口中还提供了更直接的方法，可以通过菜单对一些图形属性直接进行设置。

在图形窗口的 Insert 菜单中可以设置一些常用的图形格式（图 6-30），这些格式包括：

• X Label（*X* 轴标识）。

• Y Label（*Y* 轴标识）。

• Z Lable（*Z* 轴标识）。

• Title（图形标题）。

• Legend（图例）。

• Colorbar（色彩条）。

• Arrow（绘制箭头线）。

• Line（绘制直线）。

- Text（书写文本）。
- Axes（创建坐标轴，在图形窗口的任意位置创建任意大小的坐标轴对象）。
- Light（创建光源，打开光源对象设置对话框）。

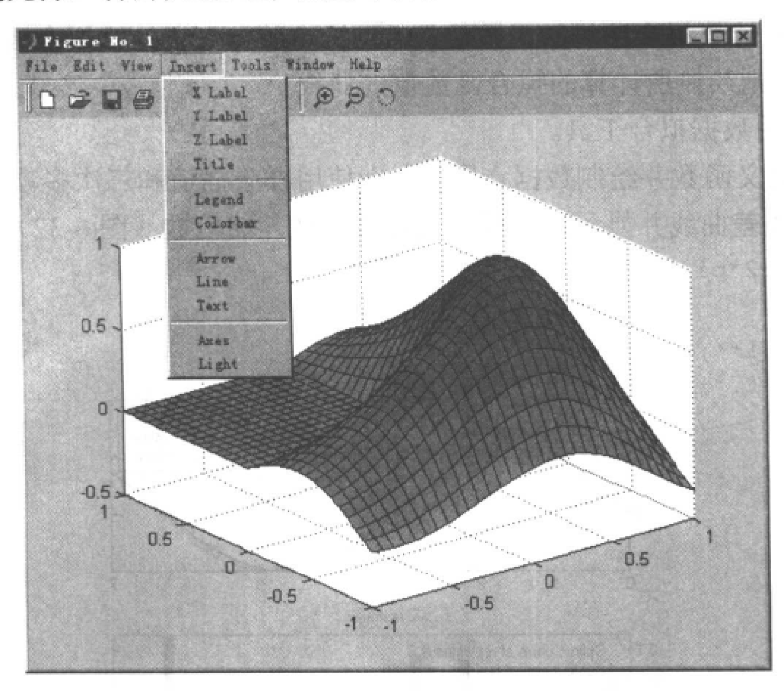


图 6-30 图形格式控制

6.4.4 数据拟合工具

MATLAB 的图形窗口中提供了功能强大的数据拟合工具。在图形窗口的 Tools 菜单中选择“Basic Fitting”，打开数据拟合工具，使用该工具可以对所绘制的曲线使用各种方法进行拟合。

数据拟合工具由 3 个窗口组成，首先打开左侧第一个窗口，按右下角的箭头可以依次打开第二个和第三个窗口，图 6-31 为全部打开的数据拟合工具窗口。

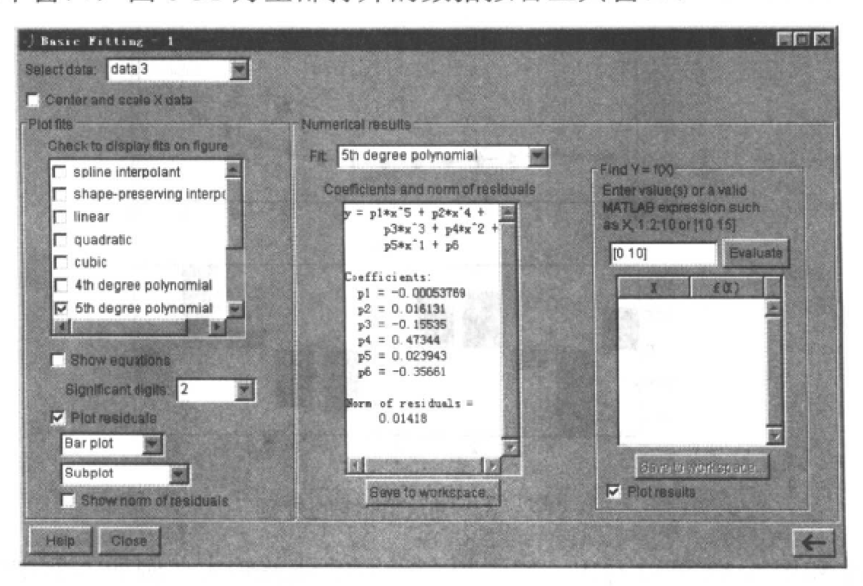


图 6-31 数据拟合工具窗口

左侧的第一个窗口中设置拟合方法、选择是否在图形上显示拟合方程及是否绘制残差图。可以选择的拟合方法包括：样条插值、保持形状插值、1 次到 10 次多项式拟合。

中间窗口用于显示拟合多项式的系数及残差的范数。

右侧窗口用于计算拟合曲线的函数值，通过给定的自变量计算拟合函数的值，如果勾选 Plot results 选项，则绘制所计算的拟合函数值的曲线。

【例 6-18】　使用数据拟合工具。

创建数据，定义函数并绘制数据点图，分别使用样条插值和三次多项式拟合方法对数据进行拟合，绘制残差曲线并显示拟合多项式系数和残差的范数（图 6-32）。

```
>> t=0:pi/5:2*pi;
>> x=sin(t);
>> plot(t,x,'*')
```

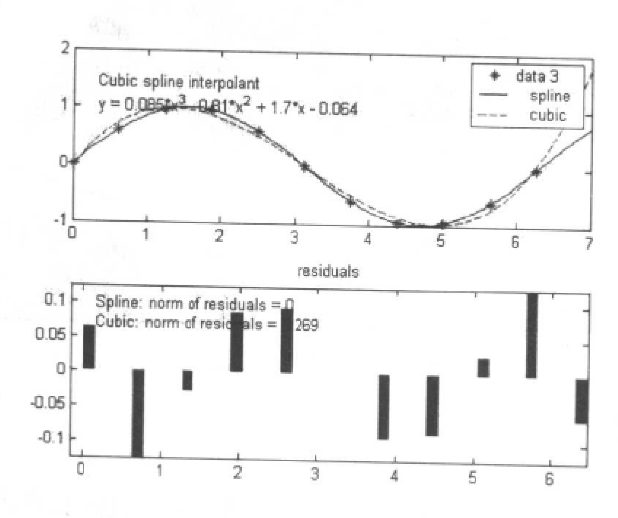


图 6-32　原始数据与拟合曲线对比

【例 6-19】　拟合曲线的外延。

使用例 6-18 的数据绘制数据点图，使用三次多项式拟合方法对数据进行拟合，计算[0: 3π]间的拟合函数曲线并绘制拟合曲线的外延图（图 6-33）。

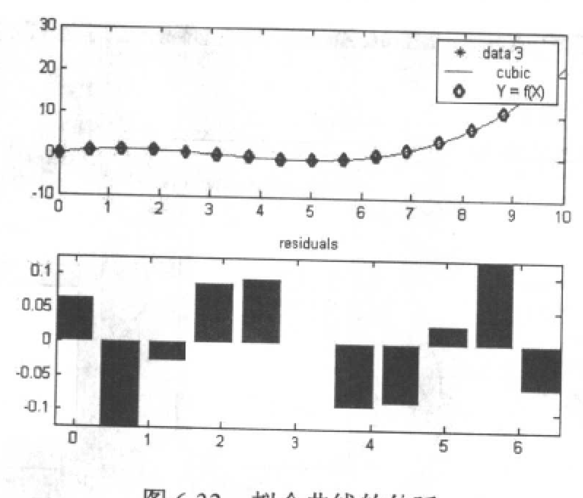


图 6-33　拟合曲线的外延

从图 6-32 和图 6-33 可以看出，如果拟合函数的残差较大（例如本例中的三次多项式），其外延曲线可能与实际曲线形成极大的偏差，这种问题在数据的拟合中要特别注意。

6.4.5 基本统计工具

MATLAB 的图形窗口中还提供了基本统计工具。在图形窗口的 Tools 菜单中选择“Data Statistics”，打开基本统计工具，使用该工具可以对所绘制的曲线进行各种基本统计运算（图 6-34）。

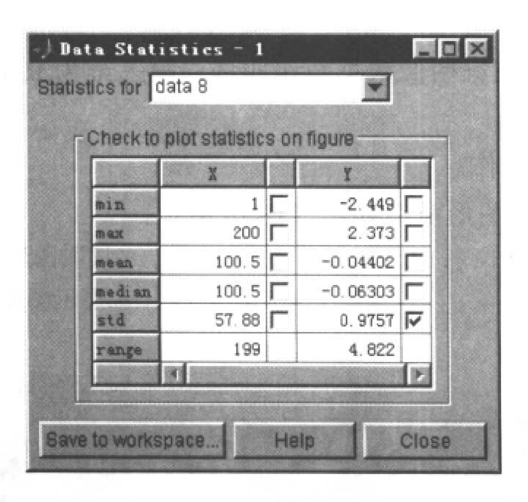


图 6-34 基本统计工具窗口

基本统计工具所提供的统计运算包括：最小值、最大值、平均值、中位值、标准差、数据范围等。图 6-35 为一组随机数据及标准差。

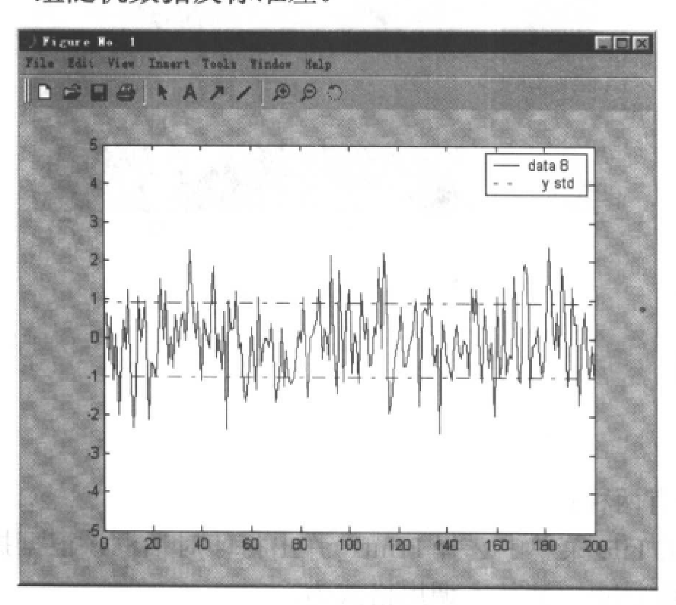


图 6-35 随机数据和标准差

6.5 其他图形处理技术

6.5.1 图像

MATLAB 中的图像分为 3 类：

(1) 索引图。由一个图像矩阵（X）和一个色彩图矩阵（MAP）组成，其调用格式为：

• [*x*, *map*] = imread('filename')

其中 filename 为 MATLAB 可以读取的图形文件格式，如.bmp、.jpg 等。

(2) 强度图（也叫灰度图）。由一个图像矩阵（*X*）和一个色彩图矩阵（*MAP*）组成，但其色彩图为单色的灰度色图，如 gray，如果有必要也可用其他色彩图，其调用格式为：

• *x* = imread('filename')

(3) 真彩色图。由一个三维图像数据数组组成，第三维为 RGB 三原色数据，没有色彩图矩阵。

【例 6-20】 图像的基本操作。

读入一个图像文件并在图形窗口显示（图 6-36）。

```
>> [x,map]=imread('lily.tif');          %读取图形文件
>> image(x);                            %显示图像
>> colormap(map)
>> axis image off                       %关掉坐标轴
```

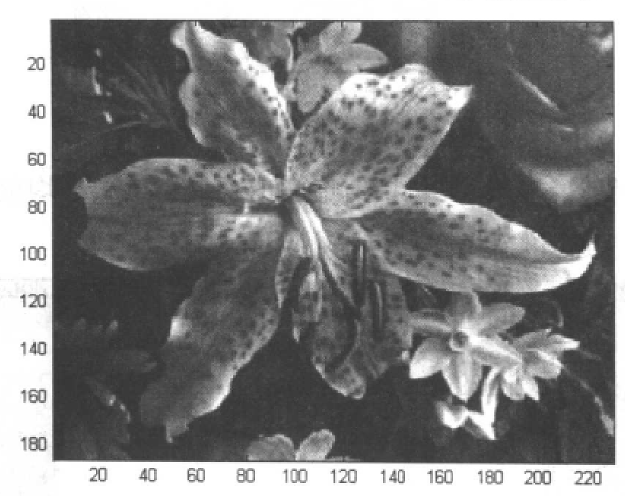


图 6-36　图像基本操作

6.5.2　图形的输出

1. 图形打印

图形打印的方法有两种：

(1) 从图形窗口的 File 菜单中选择 Print，弹出打印对话框，根据提示完成打印。

(2) 使用打印函数 print。其基本调用格式为：

• print -device -options　device 为输出设备，options 为打印选项。

【例 6-21】 图形的打印。

```
>> x=0:pi/100:2*pi;          %产生一组数据
>> plot(x,sin(x))            %绘图
>> print                     %将图形结果打印到默认打印机上
```

2. 图形存储

MATLAB 提供了多种图形存储方法。

(1) 利用 print 函数将图形输出到文件。其基本调用格式为：

• print -device -options filename　device 为输出设备，options 为打印选项，filename 为输出文件名。

【例 6-22】　建立 100 dpi 的图形文件 a01.jpg。

```
>> print(h,'-djpeg','a01','-r100')
```

(2) 生成图像文件。

图形窗口中显示的任何图形都可以通过 getframe 命令得到相应的位图数据，然后使用 imwrite 命令存储为其他格式的图像文件。

【例 6-23】　使用 imwrite 命令存储图像文件。

```
>> x=imread('flowers.tif');
>> image(x)
>> f=getframe;
>> imwrite(f.cdata,'flowers.jpg')                    % f 为结构函数
```

习　题　6

6.1　坐标轴对象是使用最多的图形对象，坐标轴对象是哪个对象的子对象，又是哪些对象的父对象？

6.2　什么是图形对象句柄？图形对象句柄有什么用途？

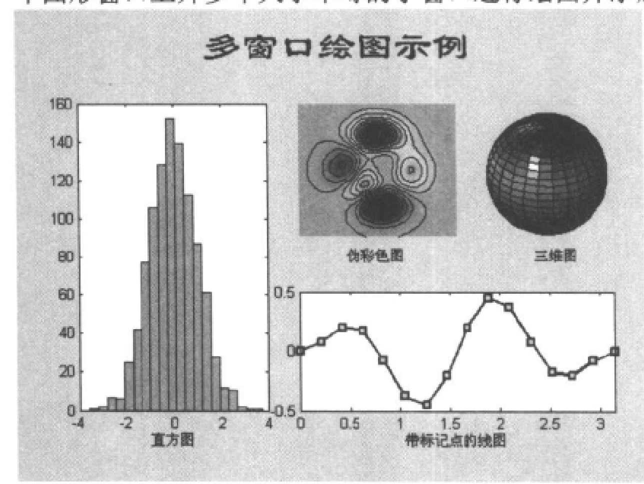
6.3　如何设置和获取指定句柄对象的属性值？假定一图形窗口对象的句柄为 h，先查询该窗口对象可以设置的各种属性，再将窗口的灰色背景设置为白色背景。

6.4　已知三维图形视角的缺省值是方位角为-37.5°、仰角为 30°，将观察点顺时针旋转 20°角应如何设置？

6.5　画一双峰曲面（peaks）图，加灯光 light，改变光源的位置并观察图形的变化。

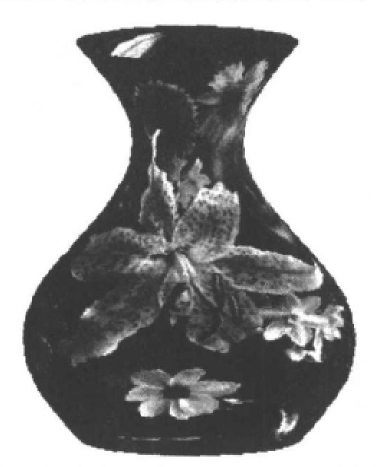
6.6　在双峰曲面上改变不同的光照模式，观察效果。

6.7　用 subplot 语句在一个图形窗口上开多个大小不等的子窗口进行绘图并添加注释，见图。



6.8　用[c, hc] = contour(peaks(30))语句绘制双峰曲面的等高线图，通过控制图形句柄的方法将第 4 条等高线加粗为 2 磅，将第 6 条等高线表示为虚线，在第 10 条等高线上加星号标记。

6.9　做一个花瓶，如图所示。（提示：做一个旋转体表面，调入一幅图像对该表面进行彩绘，即用图像的色彩图索引作为表面体的色彩图索引）



6.10　用图形窗口功能在图形窗口中的任意位置产生多个大小不等的子窗口，与 6.7 题结果进行比较。

6.11　利用图形窗口的曲线拟合工具，选择不同的拟合方法对给定的数据 y=[9,18,21,19,22,25,20,14,12,0] 进行曲线拟合，观察不同的拟合效果。

第 7 章　图形用户界面设计

随着计算机技术的飞速发展和图形化操作系统（例如 Windows）的普及，应用程序的外观发生了巨大的变化，图形用户界面（GUI）的程序已经成为应用程序的主流。使用图形用户界面的应用程序，用户可以方便地通过鼠标等输入设备与程序进行信息的交换，控制程序的运行。目前，绝大多数的软件开发工具（如 Microsoft Visual C++等）都是以图形用户界面作为应用程序开发的基础，图形用户界面的设计确定了应用程序的主要框架和基本功能，完成了窗口、图标、菜单、按钮等用户界面，软件开发者只需在由软件开发工具自动生成的程序代码中添加自己的运算或控制代码，就可以完成应用程序的设计。

MATLAB 作为功能强大的软件开发工具，提供了丰富的图形用户界面设计功能，特别是 MATLAB 6 所提供的控件种类更丰富，开发工具更完善，使用更加方便，使 MATLAB 已经具备了大型软件开发工具的所有功能和特征，用户利用 MATLAB 提供的图形用户界面设计工具，可以设计各种要求的图形用户界面，并在其基础上完成功能强大的应用程序开发。

7.1　图形用户界面开发环境

MATLAB 6.1 推出了全新的图形用户界面开发环境（GUI Development Environment，简称 GUIDE），图形用户界面开发环境提供了一组用于图形用户界面开发的工具，这些工具包括：

- 布局编辑器（Layout Editor）　在图形窗口中创建及布置图形对象。
- 几何排列工具（Alignment Tool）　调整各对象之间的相互几何关系和位置。
- 属性编辑器（Property Inspector）　查询并设置对象的属性值。
- 对象浏览器（Object Browser）　获得当前 MATLAB 窗口中图形对象句柄的分级排列。
- 菜单编辑器（Menu Editor）　建立和编辑主菜单和图形对象的鼠标右键菜单。

使用用户界面开发环境可以方便地创建 GUI 应用程序，图形用户界面开发环境根据用户设计的 GUI 布局，自动生成一个 M 文件的框架，用户使用这一框架编制自己的应用程序，在自动生成的 M 文件中采用了有效的方法管理对象句柄、执行调用函数等工作，提供了管理全局变量的方法，同时为调用函数自动添加子函数，以利于扩大应用范围。

在 MATLAB 5 中，GUI 的设计是以 M 文件的编程形式实现的，GUI 的布局代码存储在 M 文件和 MAT 文件中。而在 MATLAB 6 中有了很大的改变，MATLAB 6 将 GUI 的布局代码存储在 FIG 文件中，同时还产生一个 M 文件用于存储调用函数，在 M 文件中不再包含 GUI 的布局代码，因此在开发应用程序时代码量大大减少。

7.1.1　布局编辑器

布局编辑器用于从控件选择板上选择控件对象并放置到布局区内，布局区被激活后就成为图形用户界面。

启动布局编辑器的方法有两种，一种是在命令窗口输入命令 guide。另一种方法是直接从启动平台上启动：在操作桌面上展开启动平台窗口中树状列表的 MATLAB 项，选择其中的 GUIDE 选项即可启动布局编辑器。这两种方法启动的布局编辑器窗口都是空的。

如果编辑一个已经存在的图形用户界面文件，键入下面的命令：

• guide *filename*　编辑文件名为 *filename* 的图形用户界面。

图 7-1 为控件选择板上带有控件名称的空白布局编辑器窗口。

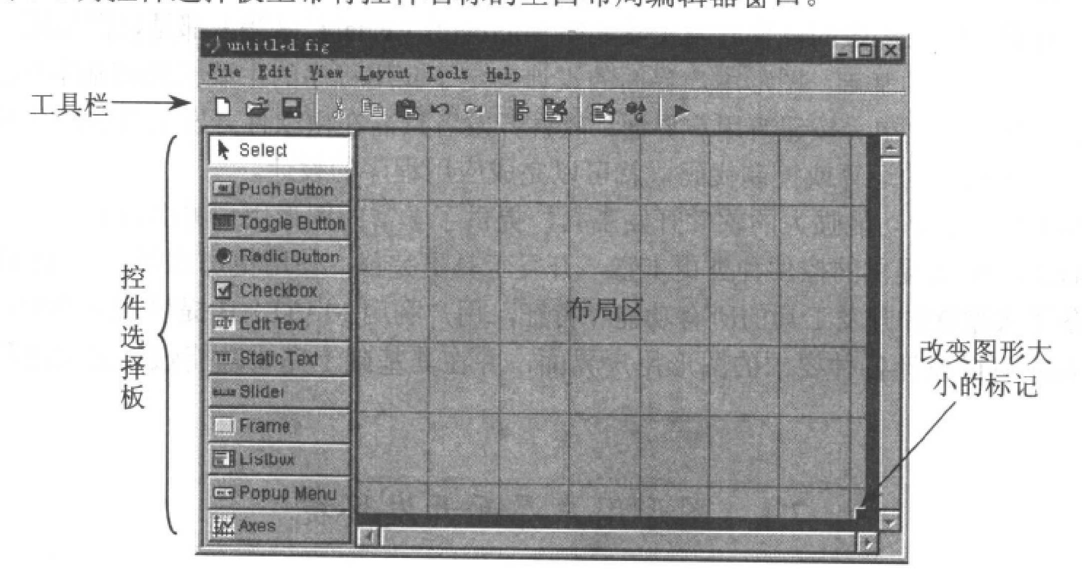


图 7-1　布局编辑器窗口

表 7-1 列出了布局编辑器窗口中工具栏的图标及功能，表 7-2 列出了布局编辑器窗口控件选择板中可选的控件。

1. 创建控件对象

使用图形用户界面开发环境创建控件对象的一般步骤如下：

(1) 鼠标点击希望创建的控件对象（以下简称控件），鼠标指针由箭头变为十字形状，移动鼠标指针到布局区希望创建控件的位置（该位置为控件图形左上角的位置），点击鼠标，即完成了控件的创建。

(2) 移动控件的位置。将鼠标指针指向要移动的控件，当鼠标指针由箭头变为十字箭头时，按住鼠标左键并拖动该控件至合适的位置，然后放开鼠标左键。

(3) 改变控件的大小。用鼠标点击希望移动的控件，控件的四个角上各出现一个小黑点，移动鼠标指针至任意一个小黑点处，当鼠标指针变为 45°方向的双箭头线时，按住鼠标左键并拖动鼠标，所出现的灰色方框随鼠标的移动而改变大小，直至满意时松开鼠标键。

(4) 选中多个对象。在布局区内空白处按住鼠标左键并拖动鼠标（不要点击控件），会随鼠标的移动拖出一个灰色的框，直至该框覆盖了所有要选中的控件时松开鼠标。所有被选中的控件的四个角都有选中标志（小黑点）。如果希望有选择地选中控件，可以按住 Ctrl

键并依次点击希望选中的控件，直至满意即可松开 Ctrl 键，所有被选中的控件可以同时移动而不改变相互间的相对位置。

表 7-1 工具栏中的图标

图标	意义
	新建
	打开
	存储
	剪切
	复制
	粘贴
	撤消
	恢复
	几何排列工具
	菜单编辑器
	属性编辑器
	对象浏览器
	激活用户界面

表 7-2 控件选择板中的控件

图标	意义
Push Button	按钮
Toggle Button	开关按钮
Radio Button	单选框
Checkbox	复选框
Edit Text	可编辑文本
Static Text	静态文本
Slider	滚动条
Frame	边框
Listbox	列表框
Popup Menu	弹出菜单
Axes	坐标轴

图 7-2 中显示了一个简单的控件布局情况：坐标轴用于显示曲线，静态文本用于说明（标题或注释），两个按钮用于控制状态（如启动、停止等），列表框用于安排一些可选择的功能，另外还有两个单选框和两个复选框用于选择某些条件等。

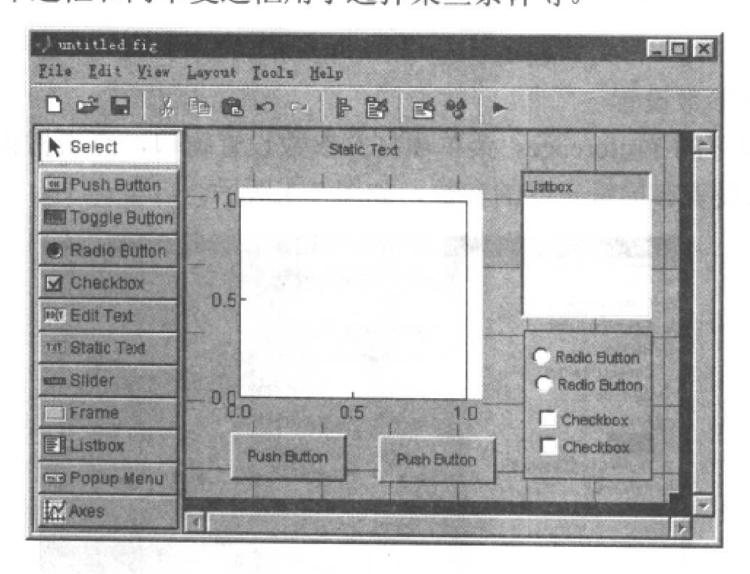


图 7-2 一个简单的布局示例

2. 激活图形用户界面

图形用户界面的创建和布局完成后，所有的控件还只存在于布局编辑器中，没有成为可以运行的图形用户界面。选择 Tools 菜单中的“Activate Figure”项或点击工具条上的“Activate Figure”按钮，出现以下情况：

图形用户界面开发环境第一次存储时同时存储 M 文件和 FIG 文件，如果所建立的布局

还没有进行存储，用户界面开发环境将打开一个“Save As”对话框，按照输入的文件的名字，存储一对同名的 M 文件和带有.fig 扩展名的 FIG 文件，当存在同名的 M 文件时，用户界面开发环境将提醒你是否覆盖或添加已存在的 M 文件。

3. 运行图形用户界面程序

MATLAB 中使用 openfig、open 或 hgload 命令调入图形用户界面程序，这些命令将 FIG 文件调入 MATLAB 的工作空间。也可以直接执行由图形用户界面开发环境生成的 M 文件。

例如，图 7-13 的程序已存储为 guitext1.m，显示该程序图形用户界面的方法为：

(1) 执行 M 文件。直接键入文件名 guitext1。

(2) 打开图形文件拷贝。openfig guitext1。

两种方法都能得到图 7-3 的结果。

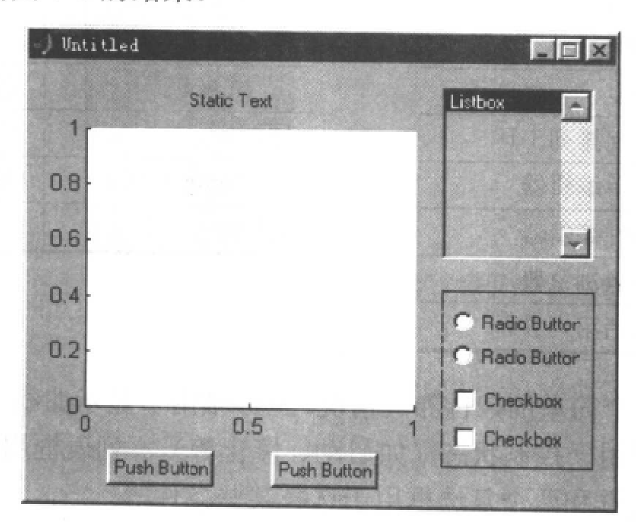


图 7-3　运行图形用户界面程序

4. 布局编辑器参数设置

选择 File 菜单下的 Preferences 菜单项打开参数设置窗口，点击树状目录中的用户界面开发环境项，可以设置布局编辑器的参数，如图 7-4 所示。

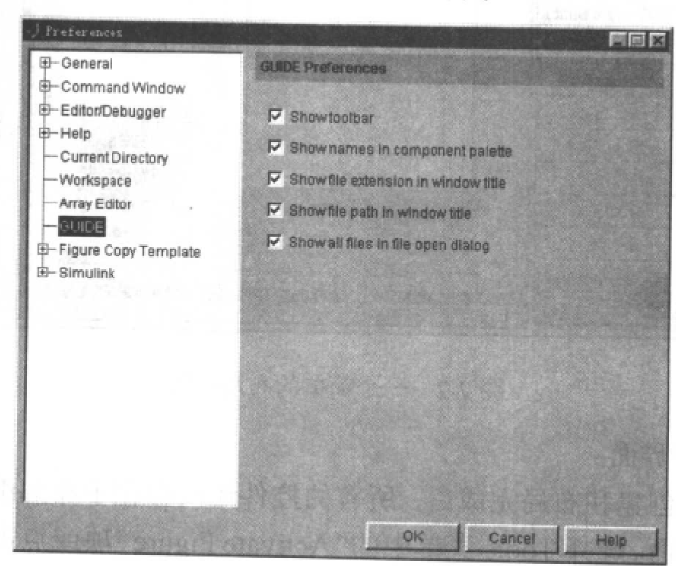


图 7-4　用户界面开发环境参数设置窗口

在用户界面开发环境的参数设置窗口中共有 5 个用于改变显示状态的复选框，通过是否勾选来确定是否选择这些参数。

(1) Show toolbar（显示工具条）。

(2) Show names in component palette（在控件选择板中显示控件名称）。

(3) Show file extension in window title（在布局编辑器窗口的标题栏中显示文件名）。

(4) Show file path in window title（在布局编辑器窗口的标题栏中显示文件路径）。

(5) Show all file in file open dialog（在打开文件的对话框中显示所有文件）。

5. 布局编辑器的弹出菜单

在布局编辑器中的布局区或任一控件上按鼠标右键，会弹出一个菜单，通过该菜单可以完成布局编辑器的大部分操作（图 7-5）。

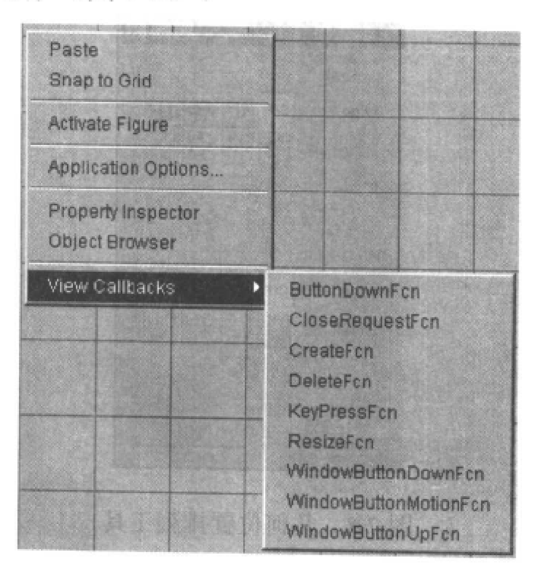


图 7-5　布局编辑器弹出菜单（在布局区不选择控件时）

如果选择了控件，菜单的内容会发生一些变化，按钮控件的弹出菜单见图 7-6。

选择不同的控件时，控件的性质决定了弹出菜单的内容可能不同，例如一般的按钮、列表框、滚动条、选择框等控件都要响应鼠标的动作，因此在弹出菜单中有 callback 菜单项，而对于坐标轴、边框和静态文本等无响应操作的控件，弹出菜单中的 callback 菜单项不激活，坐标轴控件的弹出菜单见图 7-7。

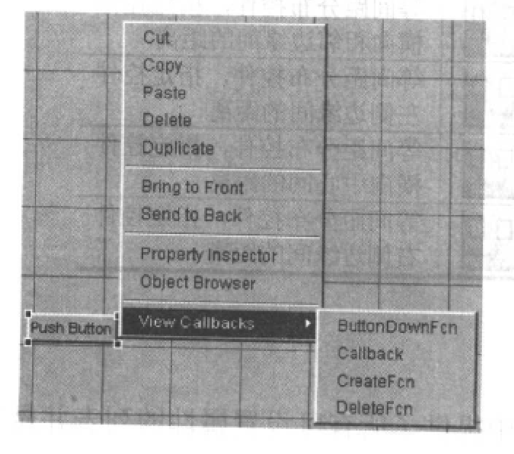


图 7-6　布局编辑器弹出菜单（选中按钮控件时）

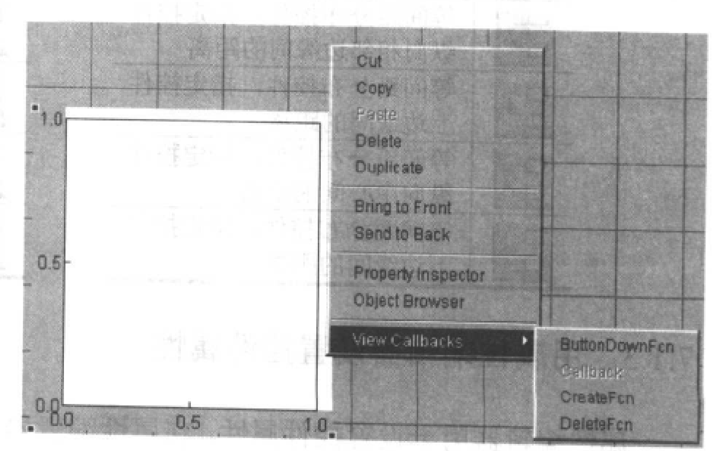


图 7-7　布局编辑器下拉菜单（选中坐标轴控件时）

弹出菜单与主菜单及工具栏按钮的功能基本相同（参见表 7-1）。

7.1.2　几何位置排列工具

几何位置排列工具（简称为排列工具）用于调节各控件对象之间的相对位置。选择工具栏中的排列工具图标即可打开排列工具窗口，见图 7-8。

排列工具分为对控件对象垂直和水平两个方向几何位置的排列，每个方向又分为位置调整和分布调整。

表 7-3 列出了排列工具中用于调整控件垂直位置的图标及功能，表 7-4 列出了排列工具中用于调整控件水平位置的图标及功能。

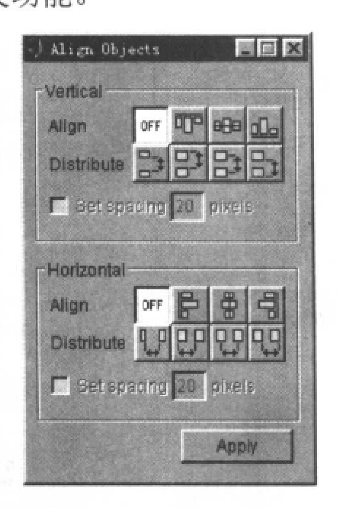


图 7-8　几何位置排列工具

表 7-3　垂直位置调整

图标	功能
	按上边缘对齐调整控件
	按中心对齐调整控件
	按下边缘对齐调整控件
	等间距分布控件，指定控件纵向相邻边缘间的距离
	等间距分布控件，指定控件上边缘间的距离
	等间距分布控件，指定控件纵向中心间的距离
	等间距分布控件，指定控件下边缘间的距离

表 7-4　水平位置调整

图标	功能
	按左边缘对齐调整控件
	按中心对齐调整控件
	按右边缘对齐调整控件
	等间距分布控件，指定控件横向相邻边缘间的距离
	等间距分布控件，指定控件左侧边缘间的距离
	等间距分布控件，指定控件横向中心间的距离
	等间距分布控件，指定控件右侧边缘间的距离

7.1.3　用属性编辑器设置控件属性

属性编辑器用于设置控件属性，在属性编辑器中提供了所有可设置属性的列表并显示出当前的属性值，通过修改属性值完成对各种属性的修改。

1. 打开属性编辑器

有 3 种途径可以打开控件对象的属性编辑器：(1) 用工具栏上的图标打开；(2) 从 View 菜单中选择 Property Inspector 菜单项；(3) 在鼠标右键弹出菜单中选择 Property Inspector 菜单项。图 7-9 显示的是按钮对象的属性编辑器。

属性编辑器左边的树状列表显示所有属性名称，右边显示对应的属性值，中间的图标是一些相关的工具（如用于调整颜色的调色板、用于输入文本的文本窗口、用于显示和修改数据的数组编辑器等）。

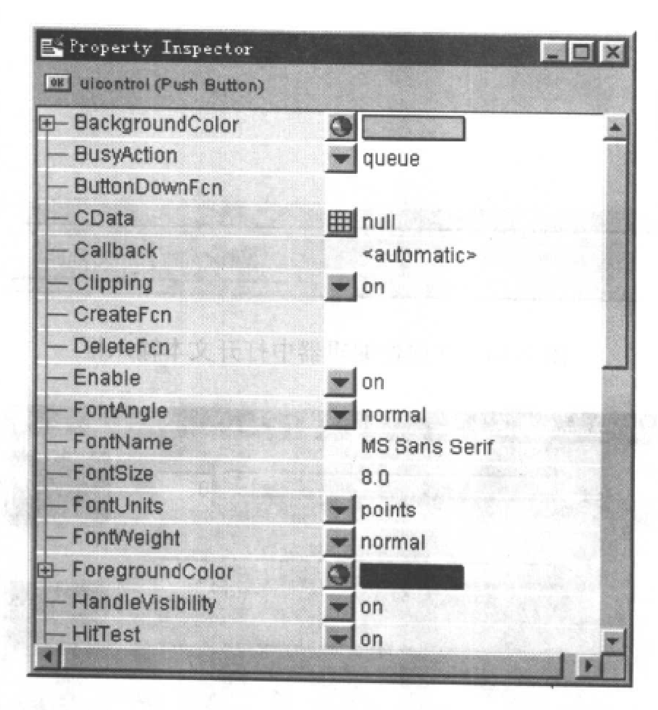


图 7-9 按钮对象的属性编辑器

2. 属性编辑器的使用

下面通过一个简单的例子说明属性编辑器的使用。

(1) 创建控件。在布局编辑器中创建控件，本例中创建了一个用于绘图的坐标轴控件、一个显示标题的文本框、一个选择三维旋转的复选框和一个启动按钮。

文本框用于显示标题，因此放在图形窗口的上方，并适当拉大。坐标轴用于显示图形，放在图形窗口的中部，为了清楚地显示图形，在不影响其他控件位置的情况下尽量拉大坐标轴的显示区。按照习惯将启动按钮放在图形窗口的右下角，考虑整体效果，复选框放在按钮的附近。

(2) 定义文本框的属性。选中布局区中的文本框并打开属性编辑器，由于该文本框用于显示标题，因此需要输入文字、定义字体和颜色。

首先点击 String 右侧的属性值区域并输入标题内容“简单设计示例”。如果输入的内容较多，可点击中间的文本输入框图标，打开一个文本输入窗口（图 7-10）。

设置字体：点击 FontName 行右侧的输入区输入“隶书”，将标题字体定义为隶书。点击 FontSize 右侧的输入区将标题字体的尺寸定义为 22。

设置颜色：点击 ForeGroundColor 行上的调色板图标，打开调色板（图 7-11）。

在调色板中通过滑条调节 R、G、B 三个颜色的值（也可以在数字输入框中直接输入颜色值），分别显示调整前与调整后的颜色，本例中选择为蓝色。

(3) 定义坐标轴的属性。本例中不对坐标系做特别的定义，只是将 Visible 属性设置为 off，目的是打开图形窗口时先不显示坐标轴，只有当点击启动按钮执行绘图时才显示坐标轴。



图 7-10　在属性编辑器中打开文本输入框

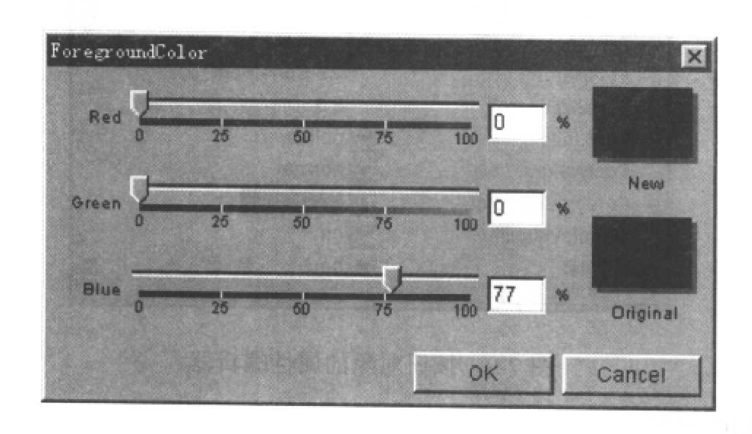


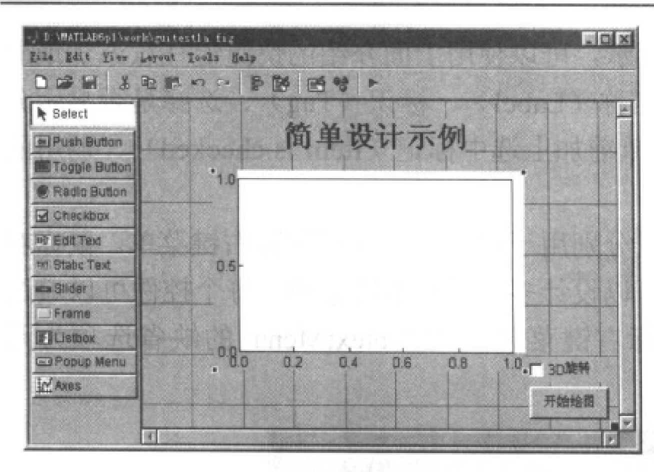
图 7-11　调色板

(4) 定义按钮的属性。定义按钮外形的方法与前面定义标题方法相似，可以设置按钮的名称（String 属性）、字体（FontName 属性）、字的颜色（ForegroundColor 属性）、尺寸（FontSize 属性）、按钮颜色（BackgroundColor）等。

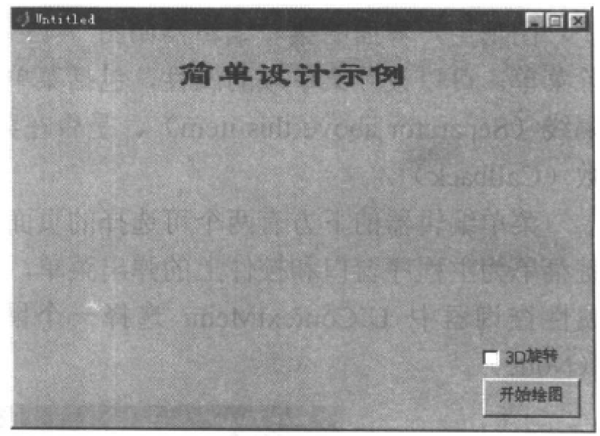
由于按钮是动作控件，因此必须定义回调函数（Callback），Callback 可以执行语句，也可以调用函数。本例中点击按钮执行三维绘图，因此在 Callback 右面的输入区输入命令：surf(peaks(30)，如果 Callback 调用的代码较多，可以在自动建立的 M 文件的 Callback 函数中添加代码。

(5) 定义复选框。首先用 String 属性定义复选框控件的标题“3D 旋转”。定义回调函数 Callback 为 rotate3d，当勾选了该复选框后，可以用鼠标控制图形进行三维旋转。

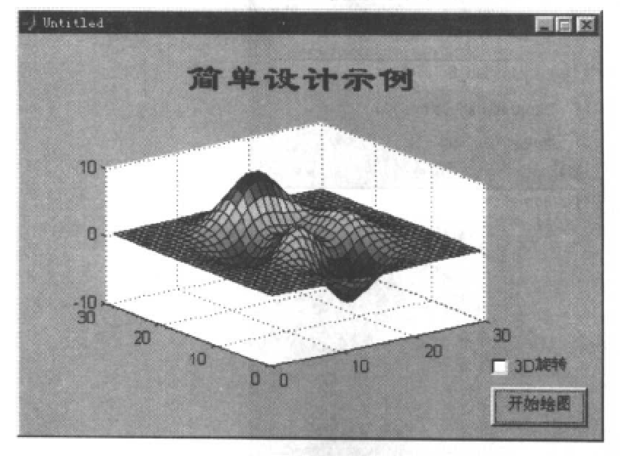
图 7-12(a) 为布局编辑器中编辑完成的图形用户界面。图 7-12(b) 为运行本例所创建的图形用户界面程序。图 7-12(c) 为点击按钮的结果，图 7-12(d) 为选择三维旋转功能后对图形进行的旋转操作。



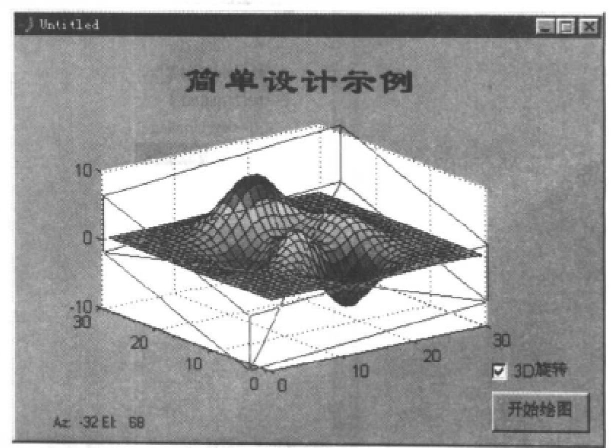
(a) 布局窗口



(b) 运行后的程序



(c) 点击开始绘图按钮



(d) 选择三维旋转功能

图 7-12 简单图形用户界面的设计与运行

7.1.4 菜单编辑器

图形用户界面的菜单编辑器包括菜单的设计和编辑。

菜单编辑器有 8 个快捷键，其功能见表 7-5。

表 7-5 菜单编辑器的快捷键

图标	功能
	创建新的菜单项
	创建新的子菜单项
	创建新的鼠标右键菜单
	变为上一级菜单
	变为下一级菜单
	菜单项上移
	菜单项下移
	删除菜单项

图形用户界面的菜单编辑器如图 7-13 所示，可以使用上面介绍的快捷键任意添加或删除菜单。可以设置菜单项的属性，包括菜单名称（Label）、标识（Tag）、选择是否显示分隔线（Separator above this item）、是否在菜单前加上选中标记（Item is checked）、调用函数（Callback）。

菜单编辑器的下方有两个可选择的页面，分别用于设计主菜单和鼠标右键菜单。鼠标右键菜单为主程序窗口和控件上的弹出菜单，可以设计多个鼠标右键菜单，每个控件可以通过属性查询器中 UIContextMenu 选择一个鼠标右键菜单，UIContextMenu 的缺省选项为无（None）。

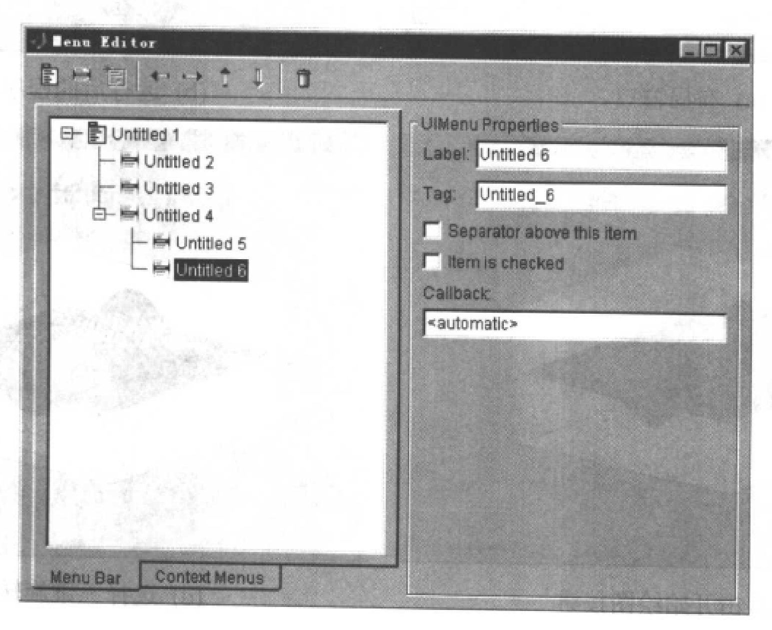


图 7-13　菜单编辑器

7.1.5　对象浏览器

对象浏览器用于浏览当前图形用户界面程序中所使用的全部对象信息（图 7-14）。在对象浏览器中用图标表示控件的类型，同时显示控件的名称或标识，在控件上双击鼠标打开该控件的属性编辑器。可以在对象浏览器中选中一个或多个控件，在对象浏览器中选中控件与在布局区直接选择控件方法的结果是一致的。

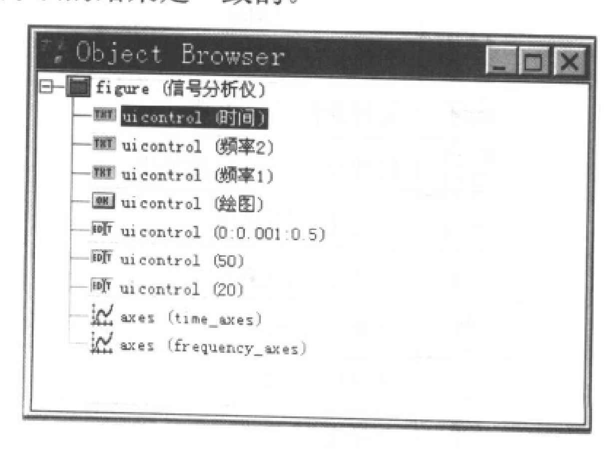


图 7-14　对象浏览器

7.2 控件对象的属性

控件对象的属性包括公共属性、基本控制属性、修饰控制属性、辅助属性和 Callback 管理属性几类，这些属性在表 7-6~表 7-10 中列出。

表 7-6 控件对象的公共属性

属性名称	含义
Children	取值为空矩阵，因为控件对象没有自己的子对象
Parent	取值为某个图形窗口对象的句柄，该句柄表明了控件对象所在的图形窗口
Tag	取值为字符串，定义了控件的标识值，在任何程序中都可以通过这个标识值控制该控件对象
Type	取值为 uicontrol，表明图形对象的类型。对于控件对象，其类型就是 uicontrol，用户不能改写这个属性
UserData	取值为一矩阵（缺省为空矩阵）。用于保存与该控件对象相关的重要数据和信息，以达到传递数据和信息的目的。用 set 和 get 函数可以访问该属性
Visible	取值为 on（缺省值）或 off。如果该属性为 off，则控件对象在图形窗口中是不可见的，但仍然存在，可用 set 和 get 函数访问它的各个属性

表 7-7 控件对象的基本控制属性

属性名称	含义
BackgroundColor	取值为颜色的预定义字符或 RGB 数值，它定义控件对象区域的背景色，缺省颜色是系统定义的浅灰色
Callback	取值是字符串，可以是某个 M 文件名或一小段 MATLAB 语句。当用户激活控件对象（如用鼠标点击控件对象图标或移动滚动条的滑块）时，应用程序就运行该属性定义的子程序。Frames、Static Texts 和 Axes 控件对象不能定义 Callback 属性
Enable	取值为 on(缺省值)、inactive 和 off。当取值为 on 时，无论何时激活控件对象，MATLAB 都执行 Callback 属性定义的子程序，当在控件对象图标区域外 5 个像素宽的边界范围内按下鼠标时，执行由 ButtonDownFun 定义的子程序
Extend	取值为四元素矢量[0, 0, width, height]，记录控件对象标题字符的位置和尺寸，度量单位由 Units 属性定义，该属性只能读不能写
ForegroundColor	取值为颜色的预定义字符或 RGB 数值，该属性定义控件对象标题字符的颜色，缺省颜色为黑色
Max、Min	取值都为数值，缺省值分别为 1 和 0。这两个属性值对于不同的控件对象类型，其意义是不同的
String	取值为字符串矩阵或块数组，定义控件的标题或选项内容
Style	取值可以是 pushbutton（按钮，缺省值）、radiobutton（单选框）、checkbox（复选框）、edit（编辑框）、text（文字说明）、slider（滚动条）、frame（边框）、popupmenu（弹出式菜单）和 listbox（列表框），由相应的值定义控件的类型
Units	取值可以是 pixels（像素，缺省值）、normalized（相对单位）、inches（英寸）、centimeters（厘米）或 points（磅）。所有单位的度量都以图形窗口左下角为（0, 0）
Value	取值可以是矢量，也可以是数值。其含义及解释依赖于控件对象的类型

表 7-8　控件对象的修饰控制属性

属性名称	含义
FontAngle	取值是 normal（正体，缺省值）、italic（斜体）和 oblique（方头），用于定义控件对象的标题等的字体形态
FontName	取值为控件对象标题等字体的字库名，必须是系统支持的各种字库
FontSize	取值为数值，定义控件对象标题等字体的字号
FontUnits	取值为 points（磅，缺省值）、normalized（相对单位）、inches（英寸）、centimeters（厘米）或 pixels（像素）
FontWeight	取值为 normal（缺省值）、light、demi 和 bold，定义字符的粗细
HorizontalAlignment	取值是 left、center（缺省值）或 right，定义控件对象标题等的对齐方式

表 7-9　控件对象的辅助属性

属性名称	含义
ListboxTop	取值是数量值，该属性只用于 listbox 控件对象，定义了列表框中的最上方的字符串在 String 属性中的序号
SliderStep	取值是两元素矢量[minstep, maxstep]，该属性只用于 slider 控件对象
Selected	取值是 on 或 off（缺省值）
SelectionHighlight	取值是 on 或 off（缺省值），该属性决定控件对象被选中时，是否显示被选中的对象

表 7-10　Callback 管理属性

属性名称	含义
BusyAction	取值是 cancel 或 queue（缺省值），该属性决定 MATLAB 采取的控制中断执行控件对象的 callback 的调用方式
ButtonDownFun	取值是字符串，一般是某个 M 文件名或一小段 MATLAB 语句
CreateFun	取值是字符串，一般是某个 M 文件名或一小段 MATLAB 语句
DeletFun	取值是字符串，一般是某个 M 文件名或一小段 MATLAB 语句
HandleVisibility	取值为 on（缺省值）、callback 或 off，该属性定义控件对象句柄的可访问权限，决定其句柄值是否在父对象的 Children 属性中出现
Interruptible	取值是 on 或 off(缺省值)，该属性决定控件对象的 callback 是否可以被随后的 callback 调用中断

7.3　对话框设计

在图形用户界面程序设计中，对话框是重要的信息显示和获取用户输入数据的用户界面对象。使用对话框，可以使应用程序的界面更加友好，使用更加方便。MATLAB 提供了两类对话框，一类为 Windows 的公共对话框，另一类为 MATLAB 风格的专用对话框。

7.3.1　公共对话框

公共对话框是利用 Windows 资源的对话框，包括文件打开、文件保存、颜色设置、字体设置、打印设置等。

1. 文件打开对话框

文件打开对话框用于打开文件，MATLAB 函数为 uigetfile，其调用格式为：

- uigetfile　弹出文件打开对话框，列出当前目录下的所有 MATLAB 文件。
- uigetfile('FilterSpec')　弹出文件打开对话框，列出当前目录下的所有由'FilterSpec'指定类型的文件。
- uigetfile('FilterSpec', 'DialogTitle')　弹出文件打开对话框，列出当前目录下的所有由'FilterSpec'指定类型的文件，同时设置文件打开对话框的标题为'DialogTitle'。
- uigetfile('FilterSpec', 'DialogTitle', x, y)　弹出文件打开对话框，列出当前目录下的所有由'FilterSpec'指定类型的文件，同时设置文件打开对话框的标题为'DialogTitle'，x、y 参数用于确定文件打开对话框的位置。
- [*fname*, *pname*] = uigetfile(...)　返回打开文件的文件名和路径。

2. 文件保存对话框

文件保存对话框用于保存文件，MATLAB 函数为 uiputfile，其调用格式与 uigetfile 相同：

- uiputfile　弹出文件保存对话框，列出当前目录下的所有 MATLAB 文件。
- uiputfile('InitFile')　弹出文件保存对话框，列出当前目录下的所有由'InitFile'指定类型的文件。
- uiputfile('InitFile', 'DialogTitle')　弹出文件保存对话框，列出当前目录下的所有由'InitFile'指定的文件类型，同时设置文件保存对话框的标题为'DialogTitle'。
- uiputfile('InitFile', 'DialogTitle', x, y)　弹出文件保存对话框，列出当前目录下的所有由'InitFile'指定的文件类型，同时设置文件保存对话框的标题为'DialogTitle'，x、y 参数用于确定文件保存对话框的位置。
- [*fname*, *pname*] = uiputfile(...)　返回保存文件的文件名和路径。

3. 颜色设置对话框

颜色设置对话框用于图形对象颜色的交互式设置，MATLAB 函数为 uisetcolor，其调用格式为：

- c = uisetcolor(*h_or_c*, 'DialogTitle')　输入参数 *h_or_c* 可以是一个图形对象的句柄，也可以是一个三色 RGB 矢量，'DialogTitle'为颜色设置对话框的标题。

4. 字体设置对话框

字体设置对话框用于字体属性的交互式设置，MATLAB函数为uisetfont，其调用格式为：

- uisetfont　打开字体设置对话框，返回所选择字体的属性。
- uisetfont(h)　h为图形对象句柄，使用字体设置对话框重新设置该对象的字体属性。
- uisetfont(S)　S为字体属性结构变量，S中包含的属性有FontName、FontUnits、FontSize、FontWeight、FontAngle，返回重新设置的属性值。
- uisetfont(h, 'DialogTitle')　h为图形对象句柄，使用字体设置对话框重新设置该对象的字体属性，'DialogTitle'设置对话框的标题。
- uisetfont(S, 'DialogTitle')　S为字体属性结构，S中包含的属性有FontName、FontUnits、FontSize、FontWeight、FontAngle，返回重新设置的属性值，'DialogTitle'设置对话框的标题。
- S = uisetfont(...)　返回字体属性值，保存在结构变量S中。

5. 打印设置对话框

打印设置对话框用于打印页面的交互式设置，MATLAB提供了两个打印设置对话框函数，分别为pagesetupdlg和pagedlg，其中pagedlg是老版本函数，在MATLAB 6中仍然可以使用，它们的调用格式为：

- *dlg* = pagesetupdlg(*fig*)　*fig*为图形窗口的句柄，如果省略*fig*，则为当前图形窗口。
- pagedlg　设置当前图形窗口。
- pagedlg(*fig*)　设置以*fig*为句柄的图形窗口。

由pagesetupdlg函数创建的打印设置对话框见图7-15。

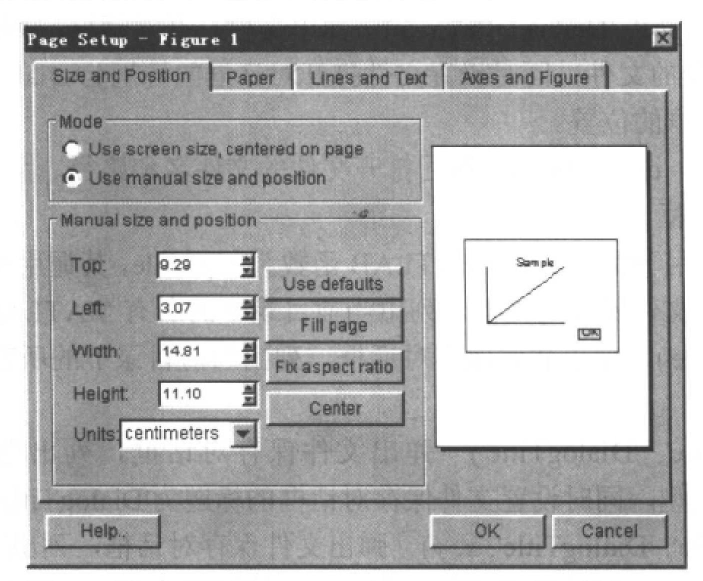


图 7-15　打印设置对话框（pagesetupdlg）

由pagedlg函数创建的打印设置对话框见图7-16。

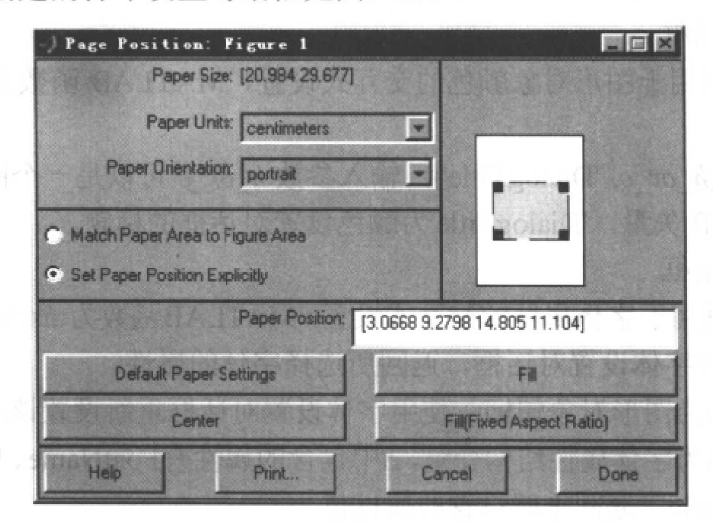


图 7-16　打印设置对话框（pagedlg）

6. 打印预览对话框

打印预览对话框用于对打印页面进行预览，MATLAB 函数为 printpreview，其调用格式为：

- printpreview　对当前图形窗口进行打印预览。
- printpreview(*f*)　对以 *f* 为句柄的图形窗口进行打印预览。

打印预览对话框见图 7-17。

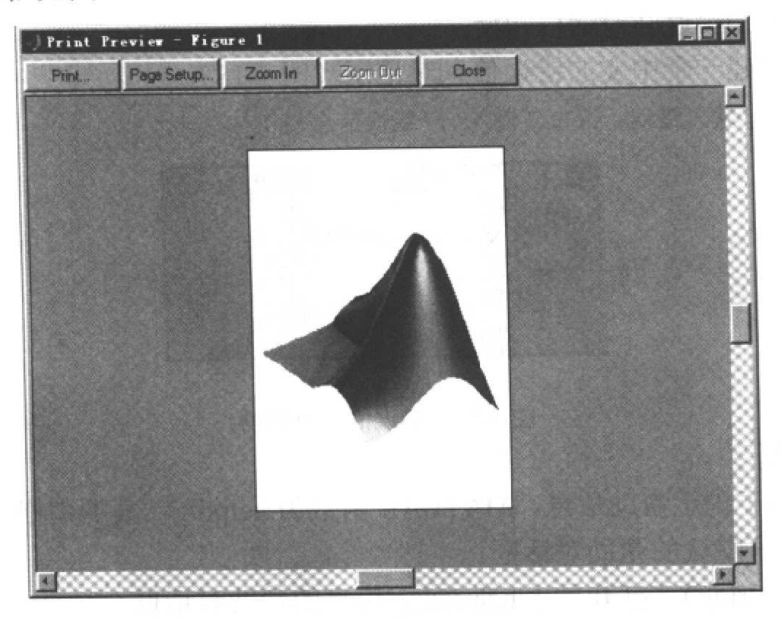


图 7-17 打印预览对话框

打印预览对话框上有 5 个按钮，分别为：

- Print（打印） 打开打印对话框。
- Page Setup（页面设置） 打开打印设置对话框，对页面进行设置。
- Zoom In（放大） 放大预览页面，超出页面的部分可使用滚动条调整。
- Zoom Out（缩小） 缩小预览页面。
- Close（关闭） 关闭打印预览对话框。

7. 打印对话框

打印对话框为 Windows 的标准对话框，MATLAB 函数为 printdlg，其调用格式为：

- printdlg 对当前图形窗口打开 Windows 打印对话框。
- printdlg(*fig*) 对以 *fig* 为句柄的图形窗口打开 Windows 打印对话框。
- printdlg('-crossplatform', *fig*) 打开 crossplatform 模式的 MATLAB 打印对话框。
- printdlg('-setup', *fig*) 在打印设置模式下强制打开打印对话框。

7.3.2 MATLAB 专用对话框

MATLAB 除了使用公共对话框外，还提供了一些专用对话框，包括帮助、错误信息、信息提示、警告信息等。

1. 错误信息对话框

错误信息对话框用于提示错误信息，MATLAB 函数为 errordlg，其调用格式为：

- errordlg 打开默认的错误信息对话框。
- errordlg('errorstring') 打开显示'errorstring'信息的错误信息对话框。
- errordlg('errorstring', 'dlgname') 打开显示'errorstring'信息的错误信息对话框，对话框的标题由'dlgname'指定。
- errordlg('errorstring', 'dlgname', 'on') 打开显示'errorstring'信息的错误信息对话框，对话

框的标题由'dlgname'指定，如果对话框已经存在，'on'参数将对话框显示在最前端。

- *h* = errordlg(...)　返回对话框句柄。

下面的例子创建一个中文的错误信息对话框（图 7-18）：

```
>> errordlg('输入错误，请重新输入','错误信息')
```

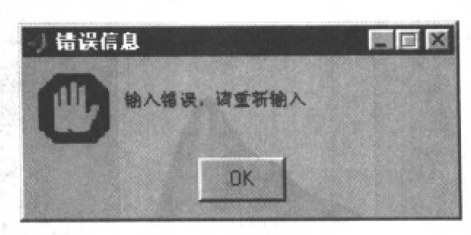


图 7-18　错误信息对话框

2. 帮助对话框

帮助对话框用于提示帮助信息，MATLAB 函数为 helpdlg，其调用格式为：

- helpdlg　打开默认的帮助对话框。
- helpdlg('helpstring')　打开显示'helpstring'信息的帮助对话框。
- helpdlg('helpstring', 'dlgname')　打开显示'helpstring'信息的信息对话框，对话框的标题由'dlgname'指定。
- *h* = helpdlg(...)　返回对话框句柄。

下面的例子创建一个中文的帮助对话框（图 7-19）：

```
>> helpdlg('矩阵尺寸必须相等','在线帮助')
```

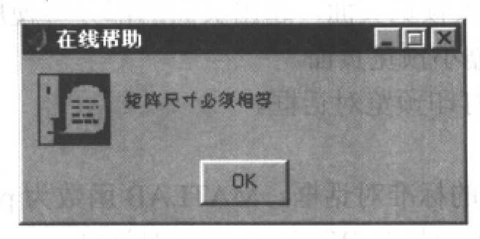


图 7-19　帮助对话框

3. 输入对话框

输入对话框用于用户输入信息，MATLAB 函数为 inputdlg，其调用格式为：

- *answer* = inputdlg(*prompt*)　打开输入对话框，*prompt* 为单元数组，用于定义输入数据窗口的个数和显示提示信息，*answer* 为用于存储输入数据的单元数组。
- *answer* = inputdlg(*prompt*, *title*)　打开输入对话框，*prompt* 为单元数组，用于定义输入数据窗口的个数和显示提示信息，*answer* 为用于存储输入数据的单元数组，*title* 确定对话框的标题。
- *answer* = inputdlg(*prompt*, *title*, *lineNo*)　*prompt* 为单元数组，用于定义输入数据窗口的个数和显示提示信息，*answer* 为用于存储输入数据的单元数组，*title* 确定对话框的标题，参数 *lineNo* 可以是标量、列矢量或 $m\times2$ 阶矩阵，如果 *lineNo* 为标量，表示每个输入窗口的行数均为 *lineNo*；如果 *lineNo* 为列矢量，则每个输入窗口的行数由列矢量 *lineNo* 的每个元素确定；如果 *lineNo* 为矩阵，每行元素对应一个输入窗口，每行的第一列为输入窗口的行数，第二列为输入窗口的宽度。
- *answer* = inputdlg(*prompt*, *title*, *lineNo*, *defAns*)　参数 *defAns* 为一个单元数组，存储每个

输入数据的默认值，元素个数必须与 *prompt* 所定义的输入窗口数相同，所有元素必须是字符串。

• *answer* = inputdlg(*prompt*, *title*, *lineNo*, *defAns*, *Resize*) 参数 *Resize* 决定输入对话框的大小能否被调整，可选的值为 on 或 off。

【例 7-1】 创建两个输入窗口的输入对话框。

定义两个输入窗口的输入对话框，第一个窗口为 2 行，第二个窗口为 1 行（图 7-20）：

```
>> prompt={'Input Name','Input Age'};
>> title='Input Name and Age';
>> lines=[2 1]';
>> def={'John Smith','35'};
>> answer=inputdlg(prompt,title,lines,def);
```

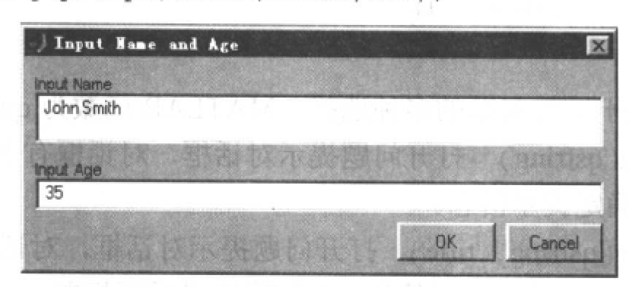


图 7-20 输入对话框

4. 列表选择对话框

列表选择对话框用于在多个选项中选择需要的值，MATLAB 函数为 listdlg，其调用格式为：

• [*Selection*, ok] = listdlg('ListString', *S*, ...) 输出参数 *Selection* 为一个矢量，存储所选择的列表项的索引号，输入参数如表 7-11 所示。

表 7-11 列表选择对话框可选的输入参数

参数	功能
'ListString'	设置列表对话框中的列表项，为字符单元数组
'SelectionMode'	通过字符串'single'或'multiple'选择单模式或多模式（缺省值）
'ListSize'	设置对话框的尺寸，由矢量[width height]确定，缺省值为[160 300]
'InitialValue'	初始选择的列表项索引，为一个矢量，缺省值为1
'Name'	设置对话框的标题，缺省值为空字符串
'PromptString'	设置对话框上面的说明文本，为字符矩阵或字符单元数组，缺省值为空
'OKString'	OK按钮的文本，缺省值为'OK'
'CancelString'	Cancel 按钮的文本，缺省值为 'Cancel'
'uh'	设置控件按钮的高度，缺省值为18
'fus'	设置框与控件对象之间的距离，缺省值为8
'ffs'	设置框与图形窗口对象之间的距离，缺省值为 8

5. 信息提示对话框

信息提示对话框用于显示提示信息，MATLAB 函数为 msgbox，其调用格式为：

• msgbox(*message*)　打开信息提示对话框，显示 *message* 信息。

• msgbox(*message*, *title*)　打开信息提示对话框，显示 *message* 信息，*title* 确定对话框标题。

• msgbox(*message*, *title*, 'icon')　打开信息提示对话框，显示 *message* 信息，*title* 确定对话框标题，'icon'用于显示图标，可以选择的图标包括：none（无图标，缺省值）、error（ ）、help（ ）、warn（ ）或 custom（用户定义）。

• msgbox(*message*, *title*, 'custom', *iconData*, *iconCmap*)　当使用用户定义图标时，*iconData* 为定义图标的图像数据，*iconCmap* 为图像的色彩图。

• msgbox(..., 'createMode')　选择模式'createMode'，选项为 modal、non-modal 和 replace。

• *h* = msgbox(...)　返回对话框句柄。

6. 问题提示对话框

问题提示对话框用于回答问题的多种选择，MATLAB 函数为 questdlg，其调用格式为：

• button = questdlg('qstring')　打开问题提示对话框，对话框有 3 个按钮，分别为 Yes、No 和 Cancel，'qstring'确定提示信息。

• button = questdlg('qstring', 'title')　打开问题提示对话框，对话框有 3 个按钮，分别为 Yes、No 和 Cancel，'qstring'确定提示信息，'title'确定对话框标题。

• button = questdlg('qstring', 'title', 'default')　当用户按回车键时，返回'default'的值，'default'必须是 Yes、No 和 Cancel 中的一个。

• button = questdlg('qstring', 'title', 'str1', 'str2', 'default')　打开问题提示对话框，对话框有两个按钮，分别由'str1'和'str2'确定，'qstring'确定提示信息，'title'确定对话框标题，'default'必须是'str1'和'str2'中的一个。

• button =questdlg('qstring', 'title', 'str1', 'str2', 'str3', 'default')　打开问题提示对话框，对话框有三个按钮，分别由'str1'、'str2'和'str3'确定，'qstring'确定提示信息，'title'确定对话框标题，'default'必须是'str1'、'str2'或'str3'中的一个。

7. 进程条

进程条以图形方式显示运算或处理的进程，MATLAB 函数为 waitbar，其调用格式为：

• *h* = waitbar(*x*, 'title')　显示以'title'为标题的进程条，*x* 为进程条的比例长度，其值必须在 0 到 1 之间，*h* 为返回的进程条对象的句柄。

• waitbar(*x*, 'title', 'CreateCancelBtn', 'button_callback')　在进程条上使用'CreateCancelBtn'参数创建一个撤消按钮，在进程中按下撤消按钮将调用'button_callback'函数。

• waitbar(..., *property_name*, *property_value*, ...)　选择其他由 *property_name* 定义的参数，参数值由 *property_value* 指定。

【例 7-2】　创建并使用进程条。

定义进程条的句柄，利用循环产生一个活动的进程条（图 7-21）：

```
>> h = waitbar(0,'Please wait...');
>> for i=1:100,
       waitbar(i/100,h)
end
```

```
>> close(h)
```



图 7-21 活动的进程条

8. 警告信息对话框

警告信息对话框用于向用户提示警告信息，MATLAB 函数为 warndlg，其调用格式为：

• h = warndlg('warningstring', 'dlgname') 打开警告信息对话框，显示'warningstring'信息，'dlgname'确定对话框标题，h 为返回的对话框句柄。

7.4 图形用户界面程序设计

本节将简单介绍如何使用图形用户界面开发环境设计应用程序。

应用程序的设计包括两个方面，首先是图形用户界面的设计，应当考虑整个图形用户界面的布局以及菜单、控件的使用和布置、事件的响应等；其次是功能的设计，通过一定的设计思路和计算方法，完成程序的设计功能。

本节将设计一个简单信号分析仪的程序，要求根据输入的两个频率和时间间隔，计算函数 $x=\sin 2\pi f_1 t+\sin 2\pi f_2 t$ 的值，并对函数进行快速傅里叶变换，最后分别绘制时域和频域的曲线。

7.4.1 设计图形用户界面

图形用户界面的设计包括以下步骤：

1. 在布局编辑器中布置控件。本例中使用了两个坐标轴、3 个文本编辑框、1 个按钮和 3 个静态文本框。

2. 使用几何位置排列工具对这些控件的位置进行调整。

3. 设置控件的属性。为显示美观，首先将文本编辑框和静态文本框的字号分别设置为 20 和 16，将 3 个静态文本框的标题分别改为“频率 1”、“频率 2”和“时间”，将按钮的标题改为“绘图”。

4. 设置其他绘图属性。如设置主窗口的标题为“信号分析仪”。

此时基本完成了图形界面的设计（图 7-22）。

7.4.2 设置控件的标识

控件的标识（Tag）用于对各控件的识别。每个控件在创建时都会由开发环境自动产生一个标识，在程序设计中，为了编辑、记忆和维护的方便，一般为控件设置一个新的标识。

在本例中设置第一个坐标轴的标识为 frequency_axes，用于显示频域图形，第二个坐标轴的标识为 time_axes，用于显示时域图形，3 个文本编辑框的标识分别为 f1_input、f2_input 和 t_input，分别用于输入两个输入频率和自变量时间的间隔，由于不需要返回 3 个静态文本

框和按钮的值，这些控件的标识可以使用缺省值（在大型程序的开发中，建议设置所有控件的标识，以利于程序的维护）。

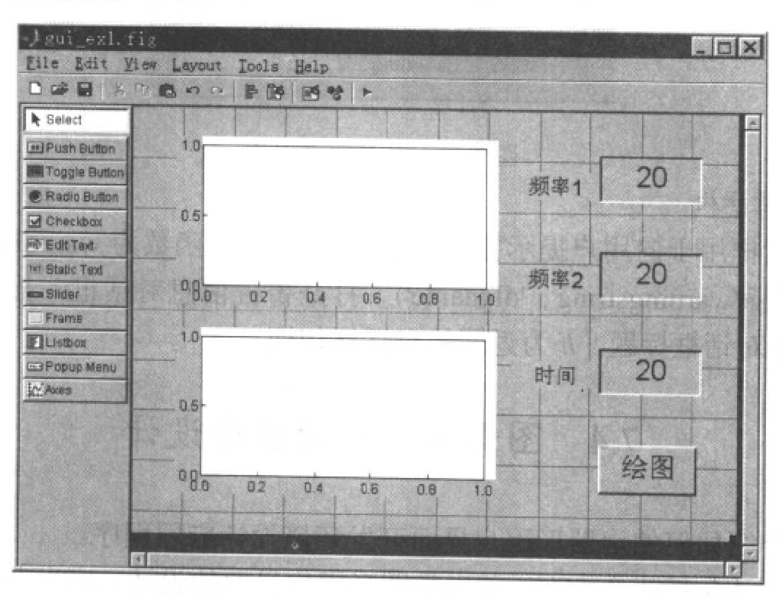


图 7-22　图形用户界面的设计

7.4.3　编写代码

为了实现程序的功能，需要编写一些代码，完成变量的赋值、输入输出、计算及绘图等工作。

1. 设置对象的初始值

分别设置 3 个文本编辑框的初始值为 f1_input=20、f2_input=50 和 t_input=0:0.001:0.5。

2. 编写代码

为按钮的调用函数编写代码，这段代码放在按钮的调用函数 pushbutton1_Callback ( ) 中，代码包括以下部分：

(1) 从图形用户界面获得用户输入的数据。本例中输入的 3 个数据分别为频率 1、频率 2 和时间间隔。

```
f1 = str2double(get(handles.f1_input,'String'));
f2 = str2double(get(handles.f2_input,'String'));
t = eval(get(handles.t_input,'String'));
```

(2) 计算数据。计算函数值，按指定点进行快速傅里叶变换，并计算频域的幅值和频域范围和分辨率。

```
x = sin(2*pi*f1*t) + sin(2*pi*f2*t);
y = fft(x,512);
m = y.*conj(y)/512;
f = 1000*(0:256)/512;
```

(3) 在第一个坐标轴中绘制频域曲线。

```
axes(handles.frequency_axes)
plot(f,m(1:257))
set(handles.frequency_axes,'XMinorTick','on')
```

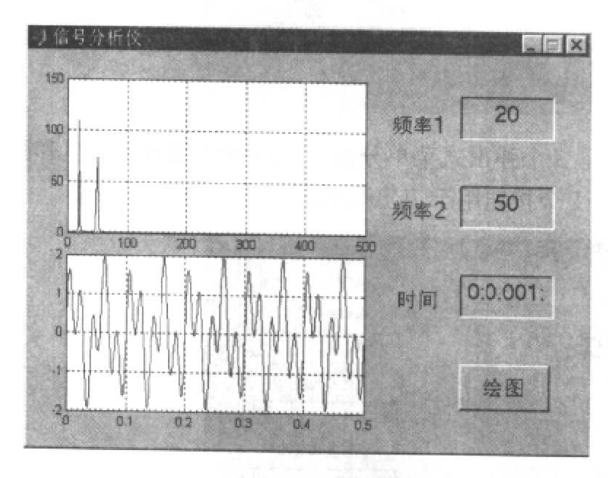
```
grid on
```

(4) 在第二个坐标轴中绘制时域曲线。

```
axes(handles.time_axes)
plot(t,x)
set(handles.time_axes,'XMinorTick','on')
grid on
```

3. 运行程序

在 3 个文本编辑框中输入不同的数值，以比较计算的结果。两组不同的输入数据的计算结果见图 7-23。



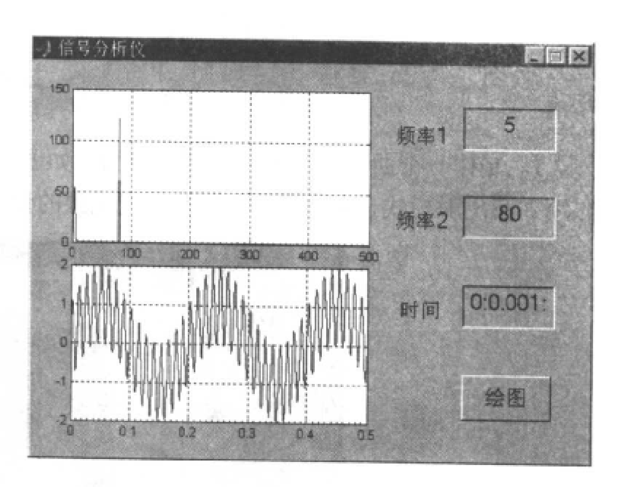
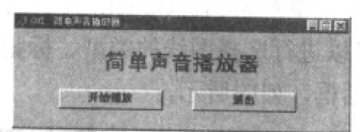


图 7-23　两组不同输入数据所得结果的对比

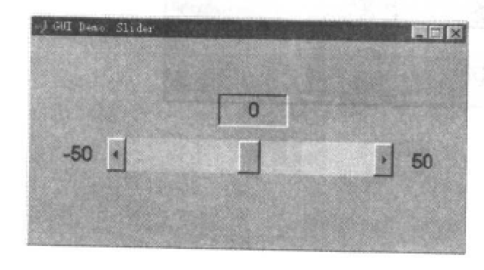
习　题　7

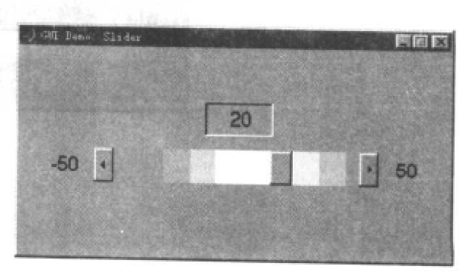
7.1　GUI 开发环境中提供了哪些方便的工具？各有什么用途？

7.2　设计一个简单的图形用户界面程序。图形界面上有一个按钮，点击按钮时，在计算机声卡中播放一段音乐。（提示，任意找一个.wav 文件，将其放在当前工作目录下或搜索路径上，当按动“开始”按钮时调入该文件并播放）

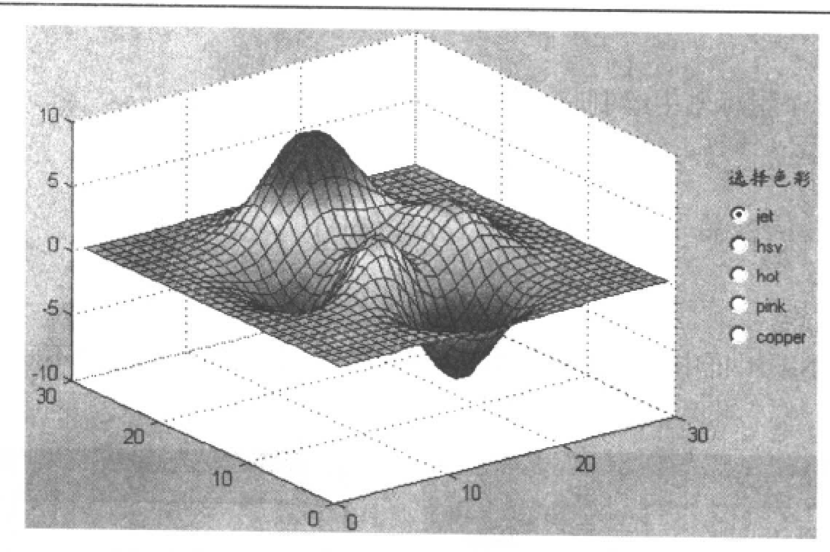


7.3　做一个滑条（滚动条）界面，图形窗口标题设置为“GUI Demo: Slider”，并关闭图形窗口的菜单条。功能：通过移动中间的滑块选择不同的取值并显示在数字框中，如果在数字框中输入指定范围内的数字，滑块将移动到相应的位置，见下图。





7.4　用单选框做一个如图所示的界面，通过选择不同的单选框来决定使用不同的色彩图。

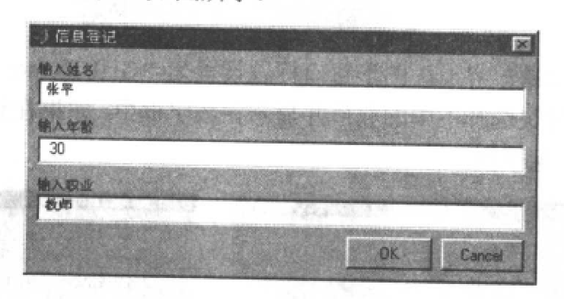


7.5　制作一个曲面光照效果的演示界面，如图所示，3 个弹出式菜单分别用于选择曲面形式、色彩图、光照模式和反射模式，3 个滚动条用于确定光源的位置，1 个按钮用于退出演示。



7.6　创建一个用于绘图参数选择的菜单对象 Plot Option，其中包含 3 个选项 LineStyle、Marker 和 Color，每个选项下面又包含若干的子项分别可以进行选择图线的类型、标记点的类型和颜色。

7.7　建立 3 个输入窗口的输入对话框，如图所示。



第8章　M文件程序设计基础

MATLAB 作为一种高级语言，不但可以以命令行的方式完成操作，而且可以像大多数高级编程语言一样具有数据结构、控制流、输入输出和面向对象编程的能力，适用于各种应用程序设计。与其他高级语言相比，MATLAB 语言具有语法相对简单、使用方便、调试容易等优点，被人们称为第四代编程语言（4GL）。

8.1　M　文　件

M 文件是由 MATLAB 命令或函数构成的文本文件，以.m 为扩展名，故称为 M 文件。M 文件有两种形式，即命令文件（Script）和函数文件（Function）。

命令文件是命令和函数的组合，执行命令文件不需要输入参数，也没有输出参数，MATLAB 自动按顺序执行命令文件中的命令，命令文件的变量保存在工作空间中。

函数文件是以 function 语句为引导的 M 文件，可以接受输入参数和返回输出参数，在缺省情况下，函数文件的内部变量是临时的局部变量，函数运行结束后，这些局部变量被释放，不再占用内存空间。用户可以根据自己的需要编制函数文件以扩充已有的 MATLAB 功能。

两种形式的 M 文件的比较见表 8-1。

表 8-1　命令文件与函数文件的比较

	命令文件	函数文件
形式	为一系列命令和函数语句的组合，不需要任何说明与定义	文件中的第一行用 function 来说明，然后再编写函数内容
参数	没有输入参数，也不用返回参数	可以接受输入参数，也可以返回参数
数据	处理的变量为工作空间变量	处理的变量为函数内部的局部变量，也可以处理全局变量
应用	自动完成一系列命令和函数，并可以多次运行。作为普通的运行程序，便于调试和修改	常用于需要反复调用并不断改变参数的场合，可用于扩充 MATLAB 函数库和一些特殊的应用
运行形式	只需在命令窗口下键入文件名即可运行	一般情况下不能靠键入文件名运行，需要由其他语句调用

8.1.1　命令文件

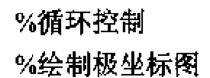
命令文件由于没有输入输出参数，是最简单的 M 文件。命令文件适用于自动执行系列 MATLAB 命令和函数，避免在命令窗口重复输入。命令文件可以调用工作空间中已有的变量或创建新的变量，命令文件结束后，这些变量仍然保存在工作空间中，直到被清除或退出

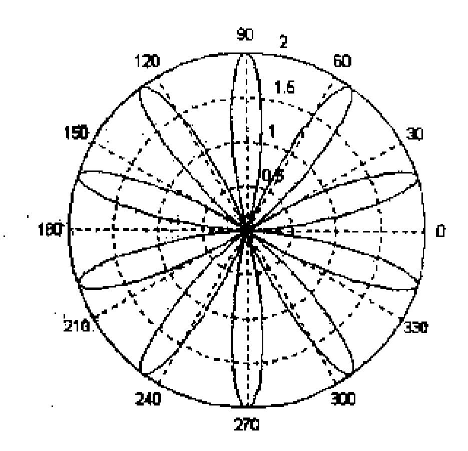
MATLAB 为止。

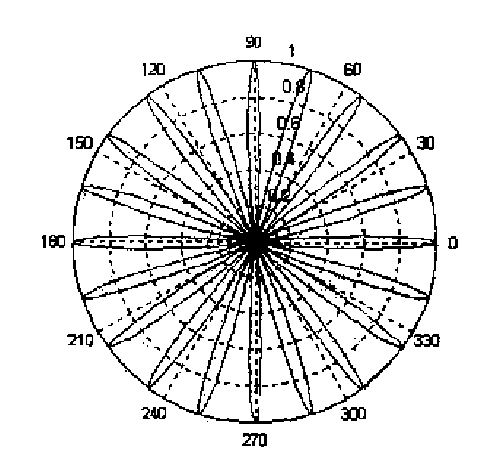
【例 8-1】　建立命令文件，绘制花瓣图。

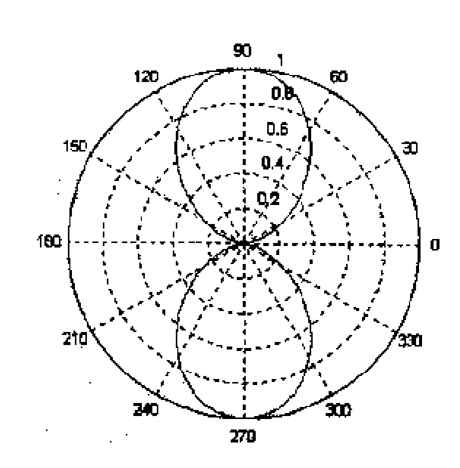
在编辑器中编辑命令，绘制图形（图 8-1）：

```
% 绘制花瓣图
theta = -pi:0.01:pi;
rho(1,:) = 2*sin(5*theta).^2;
rho(2,:) = cos(10*theta).^3;
rho(3,:) = sin(theta).^2;
rho(4,:) = 5*cos(3.5*theta).^3;
for k =1:4                                    %循环控制
   polar(theta,rho(k,:))                      %绘制极坐标图
end
```









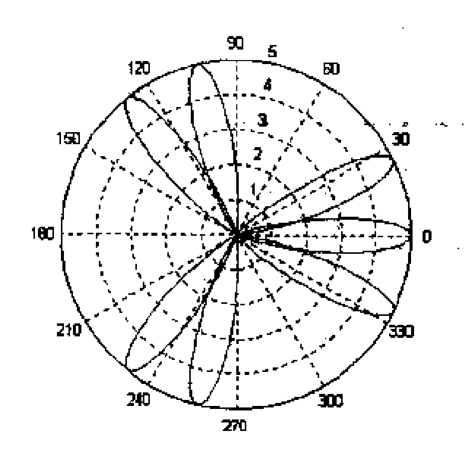


图 8-1　用命令文件绘制花瓣图

8.1.2　函数文件

函数文件可以接受输入参数和返回输出参数，函数在自己的工作空间中操作局部变量，函数的工作空间与 MATLAB 操作桌面的工作空间是不同的。

MATLAB 的函数文件由以下部分组成：

• 函数定义行　函数文件的第一行用关键字“function”把 M 文件定义为函数，并指定函数名，函数名应尽可能与 M 文件同名，同时也定义了函数的输入输出参数，例如：

function [*x*, *y*, *z*] = sphere(theta, phi, rho)

• H1 行　H1 行是帮助文本的第一行，它紧跟在定义行之后，以“%”开始，该行用于从总体上说明函数名和函数的功能，例如：

```
% AVERAGE    计算元素的平均值
```

当在 MATLAB 的命令窗口中使用 lookfor 命令查找相关的函数时，将只显示 H1 行。

• 帮助文本　帮助文本是 H1 行与函数体之间的帮助内容，也是以“%”开始，用于详细介绍函数的功能和用法以及其他说明，例如：

```
% AVERAGE(X), where X is a vector, is the mean of vector elements。
% Nonvector input results in an error。
```

• 函数体　函数体是函数的主体部分，函数体中包括该函数的全部程序代码，在函数体中可以包括流程控制、输入输出、计算、赋值、注释、图形功能以及对其他函数和命令文件的调用。

• 注释　除了函数文件开始部分的帮助文本外，可以在函数文件的任何位置添加注释语句，注释语句可以在一行的开始，也可以跟在一条可执行语句的后面（同一行中），不管在什么地方，注释语句必须以“%”开始，MATLAB 在执行 M 文件时将每一行中“%”后面的内容全部作为注释，不予以执行。

【例 8-2】　建立函数文件，计算矢量中元素的平均值。

在编辑器中编辑下列代码：

```
function y = average(x)
% AVERAGE Mean of vector elements.
% AVERAGE(X), where X is a vector, is the mean of vector elements.
% Non-vector input results in an error.
[m,n] = size(x);
if (~((m == 1) | (n == 1)) | (m == 1 & n == 1))
    error('Input must be a vector')
end
y = sum(x)./length(x);          % Actual computation
```

将上面的代码保存在文件名为 average.m 的文件中。在命令窗口生成矢量，并调用 average 函数：

```
>> z = 1:99;
>> average(z)
ans =
      50
```

8.1.3　局部变量和全局变量

通常，每一个由 M 文件定义的 MATLAB 函数都拥有自己的局部变量，这些变量独立于其他函数的局部变量和工作空间的变量，但是如果特别地定义一些变量为全局变量，这些变量就可以共享。

用关键字 global 可以把一个变量定义为全局变量。在 M 文件中定义全局变量时，如果在当前工作空间已经存在了相同的变量，系统将会给出警告，说明由于将该变量定义为全局变量，可能会使变量的值发生改变。为避免发生这种情况，应该在使用变量前先将其定义为全局变量。

在 MATLAB 中对变量名是区分大小写的，因此为了在程序中清楚表达，不发生误声明，习惯上将全局变量定义为大写字母。如：

- global A B C　将 A、B、C 三个变量定义为全局变量。

8.2　数据及数据文件

8.2.1　数据类型

MATLAB 提供了许多种数据类型用于各种运算和处理的需要，仅就数组而言就可以分为字符数组、数值数组、单元数组、结构数组、java 类和函数句柄等几大类。各种数据类型见图 8-2，各种数据类型的简要说明见表 8-2。

数组

字符 char

数值 numeric

单元 cell

结构 structure

java 类 java class

函数句柄 function handle

整数 int8,uint8, int16,uint16, int32,uint32

单精度 single

double 双精度

稀疏 sparse

用户类 user class

图 8-2　各种数据类型

在工作空间浏览器中，不同的数据类型以不同的图标表示，图 8-3 中显示了部分数据类型。

表 8-2　数据类型一览表

数据类型	举例	说明
单精度	3*10^38	单精度数值数组，单精度所需的存储空间较双精度小，但精度差，数值的范围小，单精度不能用于数学运算
双精度	3*10^300　5+6I	双精度数值数组，为最常用的MATLAB变量类型
稀疏矩阵	speye(5)	稀疏双精度矩阵，稀疏矩阵只存储少数的非零元素，较常规矩阵的存储节约了大量的存储空间，稀疏矩阵引入了特殊的算法
存储型变量	uint8(magic(3))	存储型变量为8位、16位及32位的整数数组，帮助用户有效地管理内存，实现整型变量的操作，这些变量不能用于数学运算
字符型	'Hello'	字符型数组（每个字符16位），也可以用于字符串操作

续表

单元数组	{17 'hello' eye(2)}	单元数组，单元数组的元素为其他数组，单元数组元素的尺寸、性质可以不同
结构数组	a.color = 'Red'; a.mat = magic(3);	结构数组，结构数组包括域名，域中可以包括其他数组，与单元数组类似。
用户定义类	inline('sin(x)')	MATLAB 类函数，用户定义类使用MATLAB函数创建。
Java 类	java.awt.Frame	Java 类，由Java API或第三方定义的类函数，也可以创建自己的Java类。
函数句柄	@humps	MATLAB的函数句柄，函数句柄可以在一个参数列表中传递，并使用feval运行。

Workspace

File Edit View Web Window Help

Stack: Base

Name	Size	Bytes	Class
a	50x4	1600	double array
b	3x2	48	double array (global)
c	8x1	64	double array (logical)
d	1x4	16	single array
e	1x6	12	int16 array
f	362x500x3	543000	uint8 array
g	1x1	16	function_handle array
h	1x8	16	char array
i	2x3	612	cell array
j	1x1	8648	struct array
k	1x1	140	sym object
l	1x1		java.awt.Frame
m	1x1	828	inline object
n	5x5	84	sparse array

Ready

图 8-3 在工作空间浏览器中显示的几种不同的数据类型

在图 8-3 的工作空间浏览器中，*a*、*b*、*c* 三个变量是双精度数组，其中 *b* 是全局变量，*c* 是逻辑变量，*d* 是单精度数组，*e* 是 16 位整数数组，*f* 是 8 位无符号整数数组，*g* 是函数句柄数组，*h* 是字符数组，*i* 是单元数组，*j* 是结构数组，*k* 是符号对象，*l* 是 java 类，*m* 是用户定义的类，*n* 是稀疏数组。

8.2.2 数据文件

MATLAB 允许接受的数据文件大致可以分为以下几类：

(1) 以.mat 为扩展名的二进制数据文件。.mat 文件是标准的 MATLAB 数据文件，以二进制编码形式存储，.mat 文件可以由 MATLAB 提供的 save 和 load 命令直接存取。

(2) 以.txt、.dat 等为扩展名的 ASCII 码数据文件。这种类型的数据文件也可以用 load 或 save 命令读入或存储，具体使用方法在 8.2.4 中介绍。

(3) 以.bmp、.jpg、.tif 等为扩展名的图像文件。主要用于图形图像处理。图像文件使用 imread 和 imwrite 命令读入和存储。

(4) 以.wav 为扩展名的声音文件。使用 wavread 和 wavwrite 命令读入和存储。

表 8-3 列出了 MATLAB 6.1 支持的各种数据文件及其调用方法和返回值。

表 8-3　MATLAB 6.1 中的数据文件

数据格式	文件说明	调用命令	返回值
MAT	MATLAB 工作空间变量	load	文件中的变量
CSV	以逗号分隔的数字	csvread	双精度数组
DAT	带格式的文本	importdata	双精度数组
DLM	定界的文本	dlmread	双精度数组
TAB	以 Tab 分隔的文本	dlmread	双精度数组
XLS	Excel 工作表	xlsread	双精度数组和单元数组
WK1	Lotus 123 工作表	wk1read	双精度数组和单元数组
CDF	通用数据格式	cdfread	通用数据格式的单元数组
FITS	自适应数据转换系统	fitsread	初始及扩展表数据
HDF	分层数据格式	hdfread	HDF 或 HDF-EOS 数据包
AVI	电影	aviread	MATLAB 电影
TIFF	TIFF 图像	imread	真彩色，灰度或索引图像
PNG	PNG 图像	imread	真彩色，灰度或索引图像
HDF	HDF 图像	imread	真彩色或索引图像
BMP	BMP 图像	imread	真彩色或索引图像
JPEG	JPEG 图像	imread	真彩色或灰度图像
GIF	GIF 图像	imread	索引图像
PCX	PCX 图像	imread	索引图像
XWD	XWD 图像	imread	索引图像
CUR	Cursor 图像	imread	索引图像
ICO	Icon 图像	imread	索引图像
AU	NeXT/Sun 声音	auread	声音数据及采样率
SND	NeXT/Sun 声音	auread	声音数据及采样率
WAV	Microsoft Wave 声音	wavread	声音数据及采样率

8.2.3　数据输入向导（Import Wizard）

MATLAB 操作桌面的 File 菜单下有一个非常实用的工具——数据输入向导（Import Wizard），用于将各种类型数据文件中的数据导入工作空间。选择该项后弹出 Import 对话框，在对话框中输入需要导入的数据文件名后显示数据输入向导窗口，在该窗口中可以预览数据、选择变量。下面分别介绍几种不同类型数据的导入。

1. mat 数据文件

在输入文件对话框中输入一个.mat 类型的数据文件名后，数据输入向导窗口的显示如图 8-4 所示。

在窗口的上方有两个单选项，选择第一项 File，从文件中调入数据，后面的输入框用于输入文件名，也可用“Browse...”打开文件浏览器以选择输入文件。如果选择第二项 Clipboard，则从剪贴板中调入数据。

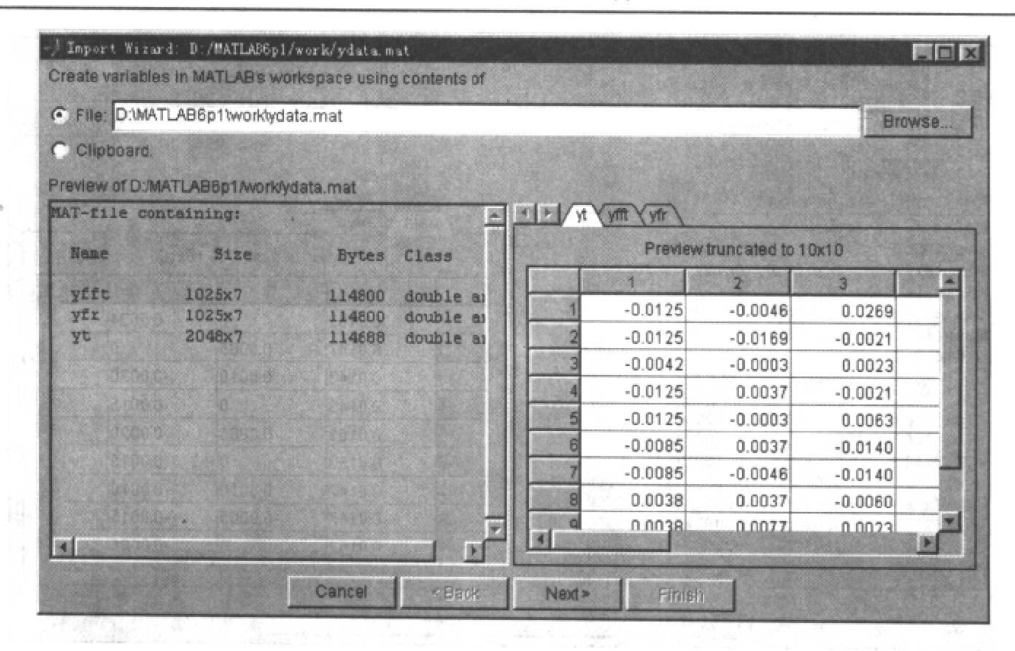


图 8-4 导入.mat 数据文件时数据输入向导窗口的第 1 页

在窗口的左侧指出该数据文件为 mat 文件，并列出变量名称、大小、类型等信息。右侧用于预览前 10 行 10 列数据，如果文件中包含多个变量，则每个变量有一个数据显示页，可以通过鼠标选择变量名标签显示相应的数据。

点击下方的 Next 按钮，显示数据输入向导窗口的第 2 页（图 8-5）。

该页的左半边显示的每个变量名前都有一个复选框，通过是否勾选决定是否将该变量导入工作空间，当所有的选择完成后，按 Finish 按钮完成数据的导入操作，此时工作空间中的变量名及变量类型等与.mat 文件中的相同，按 Cancel 按钮终止数据的导入。

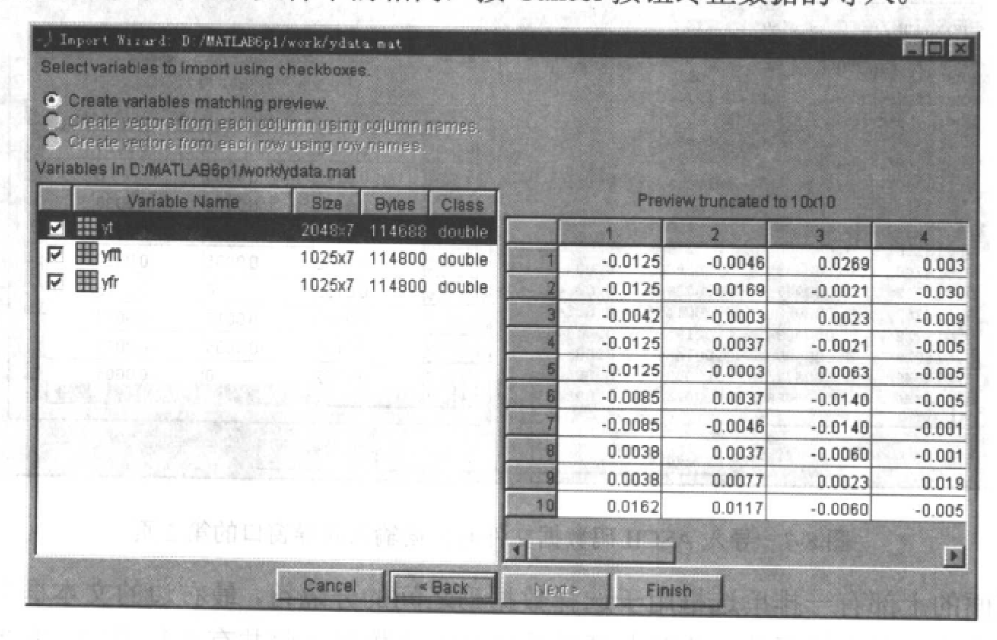


图 8-5 导入.mat 数据文件时数据输入向导窗口的第 2 页

2. ASCII 码数据文件

在输入文件对话框中输入一个 ASCII 码类型的数据文件名，数据输入向导窗口的显示如图 8-6 所示。

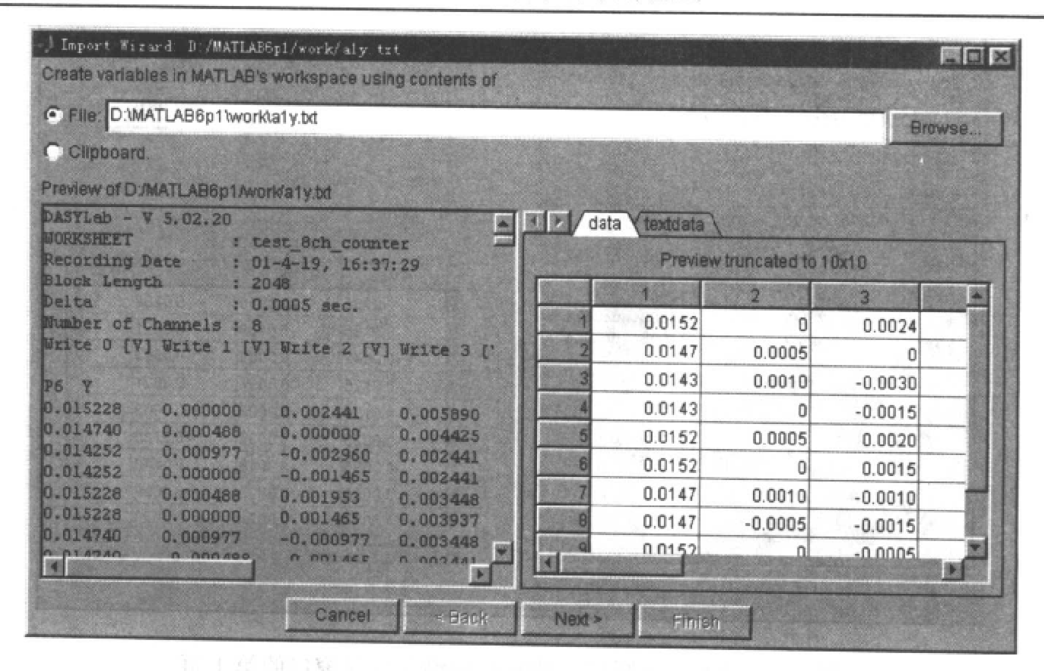


图 8-6　导入 ASCII 码数据文件时数据输入向导窗口的第 1 页

对于 ASCII 码数据，由于数据中没有变量名信息，因此在第 1 页的左侧列出了文件中的头文本信息和所有数据，右侧显示文件中前 10 行 10 列的数据。点击 Next 按钮，显示数据输入向导窗口的第 2 页（图 8-7）。

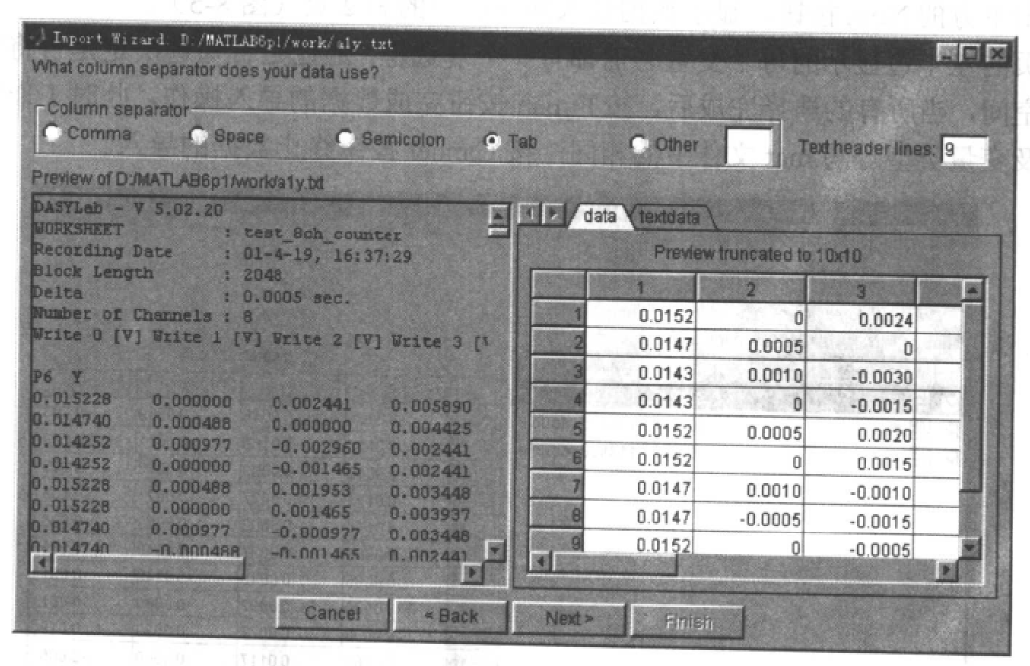


图 8-7　导入 ASCII 码数据文件时数据输入向导窗口的第 2 页

该页面的上部有一排单选框用于选择数据列之间的分隔符，最右边的文本框中显示该数据文件中头文本所占的行数，从图中可以看到在正式数据之前共有 9 行说明，右半边显示的数据从左边内容的第十行开始。在右半边的标签上可以选择显示数据内容（如图中的 data）或头文本内容（如图中的 textdata）。选择 Next 按钮，显示的窗口如图 8-8 所示。

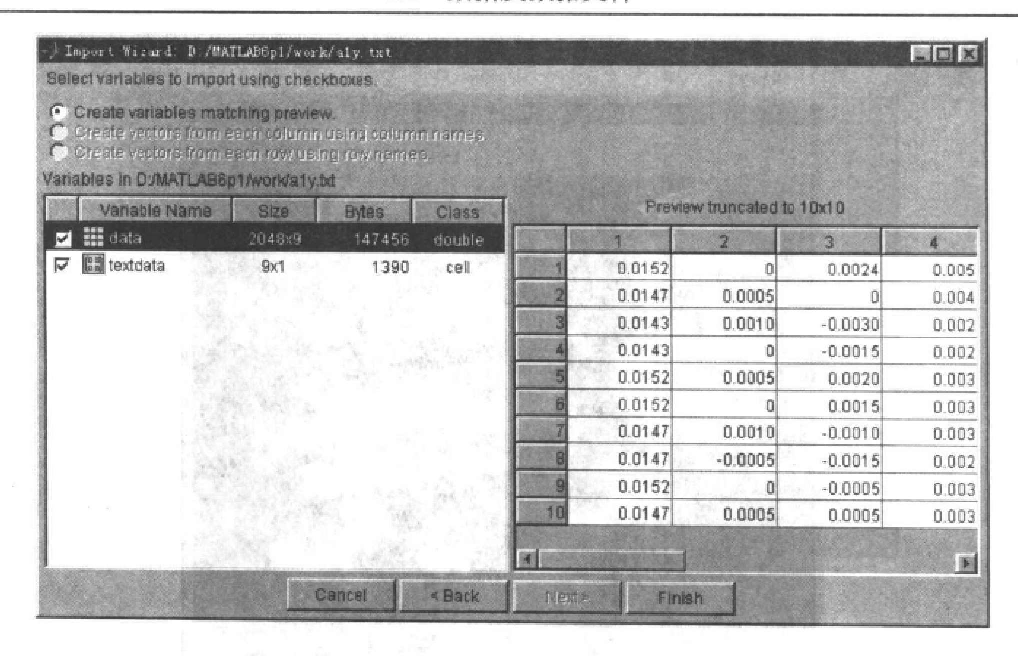


图 8-8 导入 ASCII 码数据文件时数据输入向导窗口的第 3 页

该页面用于选择导入数据的内容，对于这种带有头文本内容的 ASCII 数据文件，可以分别选择导入数据内容和头文本内容。完成所有的选择后，按 Finish 按钮完成数据的导入操作，此时导入工作空间数据的变量名为 data，头文本内容放置在变量名为 textdata 的单元数组中，按 Cancel 按钮终止数据的导入。

需要注意的是，如果导入的 ASCII 码数据文件中没有头文本内容，则存储在工作空间的变量名与文件名相同。

3. 图像数据文件

在输入文件对话框中输入一个图像文件名后，数据输入向导窗口的显示如图 8-9 所示。

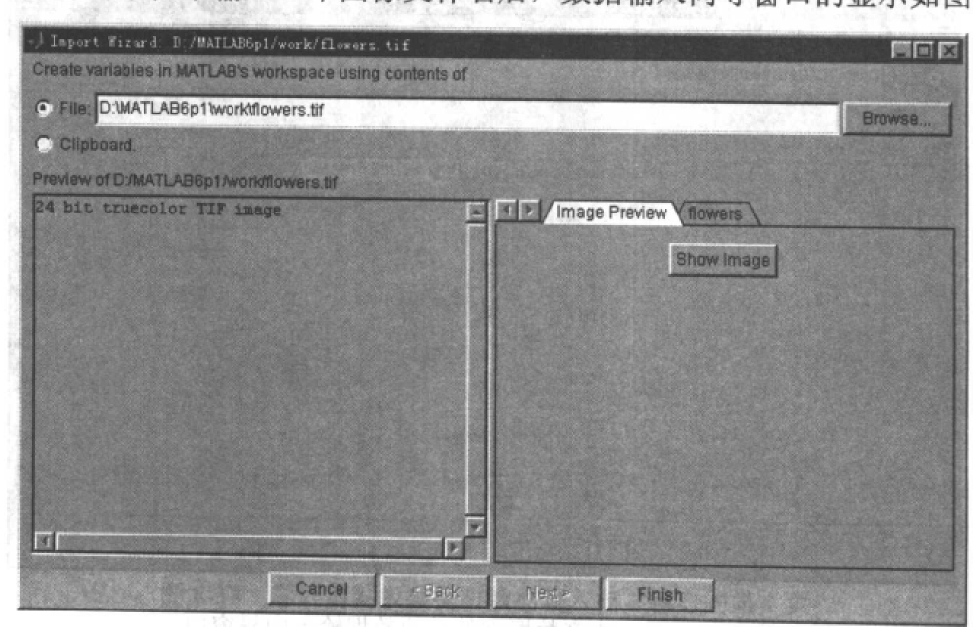


图 8-9 导入图像数据文件时的数据输入向导窗口

从图中可以看到在窗口的左侧显示该图像文件的信息：24 位真彩色 TIF 图像。右侧图像预览页面上有一个显示图像（Show Image）按钮，点击该按钮，弹出一个新的图形显示窗口

（图 8-10）。



图 8-10　图像预览窗口

在该窗口中可以用鼠标在任意位置拖曳出一个放大区域对图像进行放大。

按 Finish 按钮完成数据的导入操作，此时导入工作空间的数据变量名与文件名相同，按 Cancel 按钮终止数据的导入。

4. 声音数据文件

在输入文件对话框中输入一个以.wav 为扩展名的声音文件后，数据输入向导窗口的显示如图 8-11 所示。

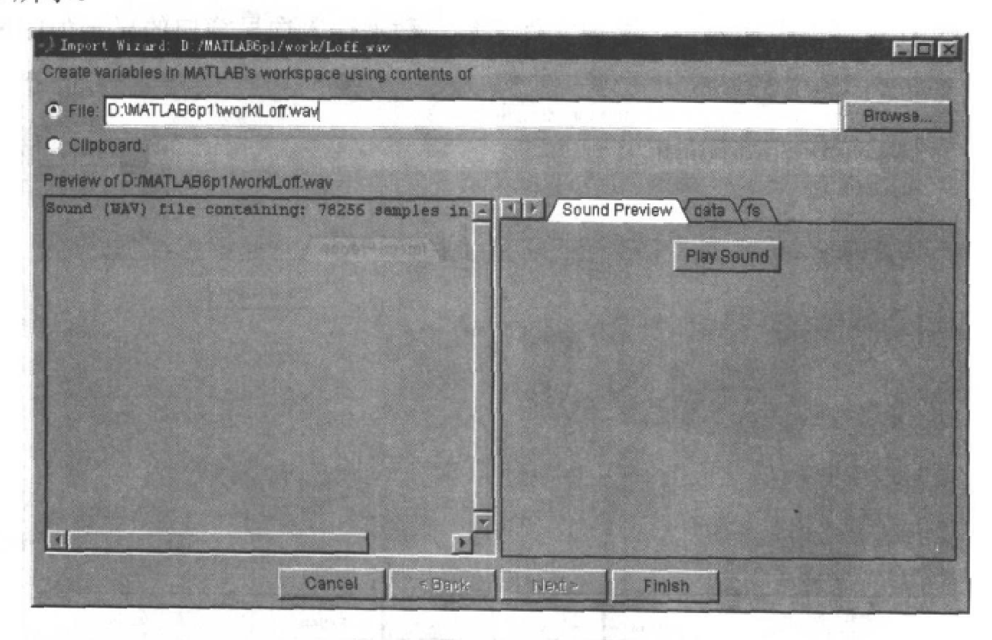


图 8-11　导入声音数据文件时的数据输入向导窗口

窗口的左侧显示该文件的信息，表明是一个 WAV 声音文件，同时显示文件的样本数和通道数。在窗口右侧的声音预览页面上有一个播放声音（Play Sound）按钮，点击该按钮，

弹出一个显示声音数据波形的图形窗口，并通过计算机的声卡播放声音（图 8-12）。

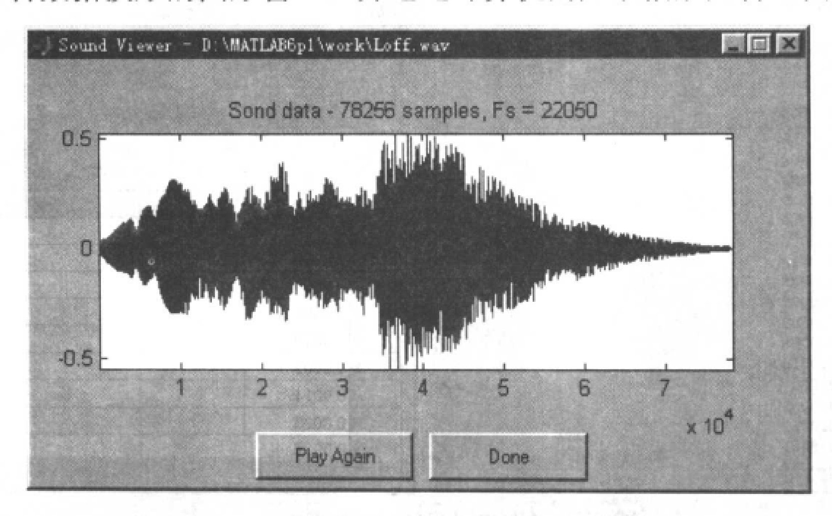


图 8-12 声音数据波形预览窗口

按“Play Again”按钮重新播放声音，按 Done 按钮关闭预览窗口。

在数据输入向导窗口的第 1 页中，除了预览页面外还有数据（data）页面和采样频率（fs）页面。选择数据页面，显示该声音文件的数字数据，选择采样频率页面显示采样频率。

按 Finish 按钮完成数据的导入操作，此时导入工作空间的数据变量名与文件名相同，变量类型为结构数组，其中包含数据和采样频率两部分，按 Cancel 按钮终止数据的导入。

5. 剪贴板的数据

如果要导入的数据已经放在系统的剪贴板中，可以在数据输入向导窗口的第 1 页中选择剪贴板单选框（图 8-13）。

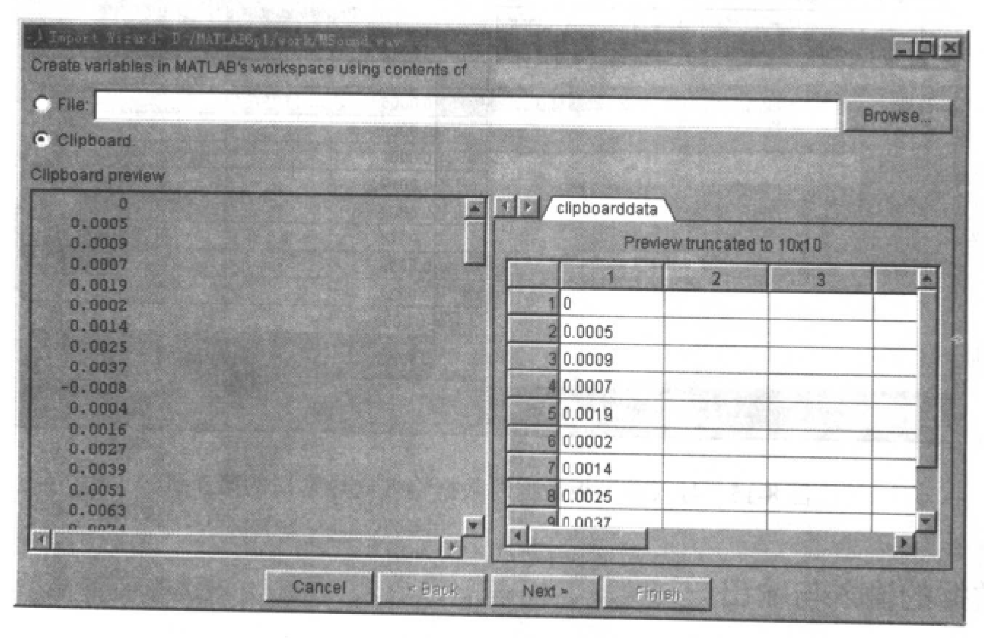


图 8-13 导入剪贴板数据时数据输入向导窗口的第 1 页

图中显示剪贴板中的数据，选择 Next 按钮，显示窗口的第 2 页（图 8-14）。

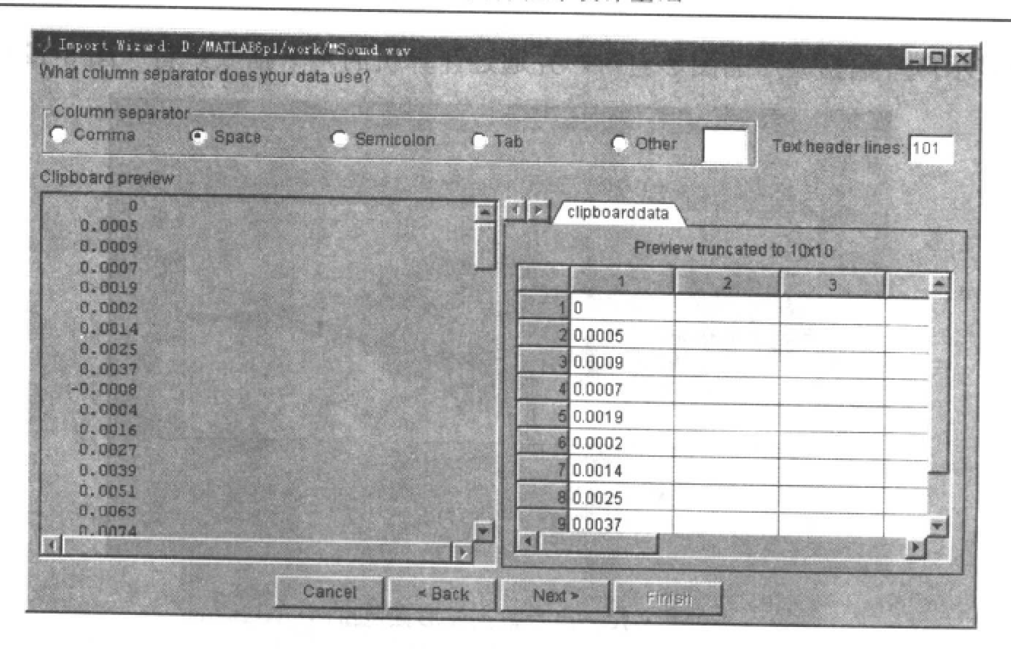


图 8-14　导入剪贴板数据时数据输入向导窗口的第 2 页

该页面可以设置行间的分隔符，选择 Next 按钮，显示窗口的第 3 页（图 8-15）。

该页面显示变量名及其数据，按 Finish 按钮完成数据的导入操作，此时导入工作空间的数据变量名为 clipboarddata，变量类型为单元数组，按 Cancel 按钮终止数据的导入。

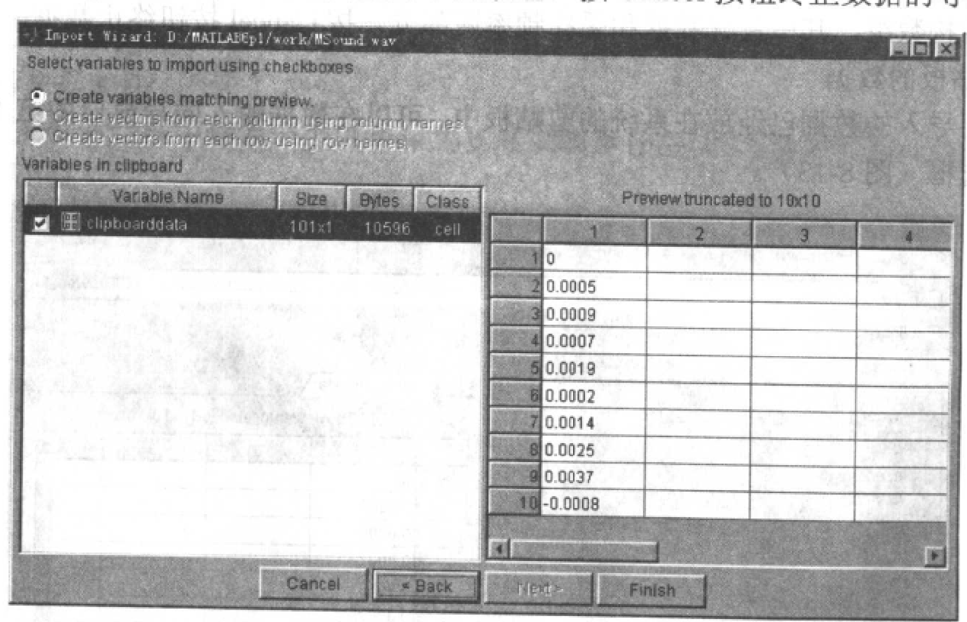


图 8-15　导入剪贴板数据时数据输入向导窗口的第 3 页

8.2.4　数据的输入与输出

1. save 的使用

save 函数将 MATLAB 工作空间中的变量存入磁盘，其调用格式为：

- save　将 MATLAB 工作空间中所有变量以二进制格式存入名为 matlab.mat 的缺省文件。

• save dfile 将工作空间中的所有变量以二进制格式存入名为 dfile.mat 的文件，扩展名自动产生。

• save dfile *x* 只把变量 *x* 以二进制格式存入 dfile.mat 文件，扩展名自动产生。

• save dfile.dat *x* –ascii 将变量 *x* 以 8 位 ASCII 码形式存入 dfile.dat 文件。

• save dfile.dat *x* –ascii –double 将变量 *x* 以 16 位 ASCII 码形式存入 dfile.dat 文件。

• save(fname, 'a', '-ascii') 此处，fname 是一个预先定义的包含文件名的字符串，该用法将变量 *a* 以 ASCII 码格式存入由 fname 定义的文件中，由于在这种用法中，文件名是一个字符变量，因此可以方便地通过编程方法存储一系列数据文件。

使用.mat 格式的文件存储数据时，可以保存当前工作空间中的各种变量。而使用-ascii 选项时，只能保存数值型变量，如果工作空间中有其他类型的变量，MATLAB 将给出警告，其他类型的变量不会被保存。如果使用-ascii 选项在一个数据文件中保存两个以上的数值型变量，这些变量将被按顺序保存在一起。因此，如果使用-ascii 选项保存数据，最好只保存一个变量或保存长度相同的多个变量。

2. load 的使用

load 函数用于将磁盘文件中的数据调入工作空间，其调用格式为：

• load 把磁盘文件 matlab.mat 的内容读入内存，由于存储.mat 文件时包含了变量名、变量类型等信息，因此回调时直接将原变量的所有信息带入工作空间。

• load dfile 把磁盘文件 dfile.mat 的内容读入内存。

• load dfile.dat 把磁盘文件 dfile.dat 的内容读入内存。由于这是一个 ASCII 码文件，系统自动将文件名（这里是 dfile）定义为变量名。

• *S* = load(*fname*) *fname* 为预先定义的包含文件名的字符串，当数据文件为.mat 文件时，load 将由 *fname* 定义文件名的数据文件调入结构变量 *S* 中，原有的变量成为结构变量 *S* 的域，如果数据文件为 ASCII 码文件，则 *S* 为数值变量，使用这种方法可以通过编程方便地调入一系列数据文件。

8.3 程序的流程控制

MATLAB 中除了按正常顺序执行程序中的命令和函数以外，还提供了 8 种控制程序流程的语句，这些语句包括 for、while、if、switch、try、continue、break、return 等。

8.3.1 循环语句

在实际问题中会遇到许多有规律地重复运算，因此在程序设计中需要将某些语句重复执行。一组被重复执行的语句称为循环体，每循环一次，都必须作出是否继续重复的决定，这个决定所依据的条件称为循环的终止条件。MATLAB 提供了两种循环方式：for-end 循环和 while-end 循环。

1. for 循环

for 语句为计数循环语句，在许多情况下，循环条件是有规律变化的，通常把循环条件的初值、判别和变化放在循环的开头，这种形式构成 for 语句的循环结构。for 循环的一般形

式为：

for v=表达式
　　语句体
end

通常情况下，表达式是一个矢量，形式为 m:s:n，其中 m 是循环初值，n 是循环终值，s 是步长，m、n、s 可以取整数、小数、正数和负数，s 的缺省值为 1。在执行 for 循环时，该矢量的元素被逐一赋给变量 v，然后执行语句体，MATLAB 的 for 循环与其他计算机语言没有区别，for 和 end 必须配对使用。

【例 8-3】 简单的 for 循环示例。

在编辑器中编辑下列命令，运行简单的 for 循环：

```
n=10;
for i=1:n
   x(i)=(i+1).^2;
end
x
x =
    4    9   16   25   36   49   64   81  100  121
```

本例中循环初值为 1，终值为 n，缺省步长为 1，循环体实现对 x 的前 n 个元素赋值。如果 n 小于 1 ，此循环仍然合法，但循环内的语句将不执行。如果 x 事先不存在或容量小于 n，那么超出的部分将自动补上。本例只是为演示 for 循环，取得本例结果还有更合理的方法。

for 循环可以嵌套，但需要注意的是，在嵌套中每一个 for 都必须与 end 相匹配，否则循环将出错。

【例 8-4】 for 循环的嵌套。

在编辑器中编辑下列命令，运行带嵌套的 for 循环：

```
m=3;
n=4;
for i=1:m
   for j=1:n
      a(i,j)=1/(i+j-1);
   end
end
format rat
a
a =
      1         1/2        1/3        1/4
    1/2         1/3        1/4        1/5
    1/3         1/4        1/5        1/6
```

【例 8-5】 运用非 1 步长 for 循环，产生 0~20 以内的偶数。

在编辑器中编辑下列命令，运行非 1 步长的 for 循环：

```
for i=0:2:20
```

```
  a(i/2+1)=i;
end
a
a =
    0   2   4   6   8   10   12   14  16  18    20
```

2. while 循环

while 语句是条件循环语句，while 循环使语句体在逻辑条件控制下重复不确定次，直到循环条件不成立为止。While 循环的一般形式是：

```
while 表达式
    语句体
end
```

只要表达式的结果非零，语句体就重复执行。当表达式的结果不是标量时，可以用 any、all 等函数处理，例如取表达式为 1 时，该循环将无休止进行。while 和 end 必须配对使用。

【例 8-6】 利用 while 循环，求解使 n! 达到 100 位数的第一个 n 是多少？

在编辑器中编辑下列命令，求解 n：

```
n=1;
while prod(1:n)<1e100
   n=n+1;
end
n
n=
   70
```

8.3.2 条件语句

在复杂的运算中常常需要判断是否满足某些条件，以选择下一步的方法和策略。一般使用条件语句完成这类判断和选择。

1. if-end 语句

if-end 语句是最简单的条件语句，其一般形式为：

```
if 表达式
    语句体
end
```

关键字 if 后的表达式确定了判断条件，表达式必须首先计算，由于 MATLAB 中没有布尔量，所以以表达式的值是否为零作为判断标准。如果表达式值不为零，则执行 if 和 end 之间的所有语句，否则跳过 if 结构，执行 end 后面的语句。

2. if-else-end 语句

if-else-end 语句在 if 和 end 之间增加一个 else 选择，语句的一般形式为：

```
if 表达式
    语句体 1;
else
```

　　语句体 2;
end

当表达式的值为真时，执行语句体 1，否则执行语句体 2。

【例 8-7】　应用 if-else-end 条件语句，判断学生是否通过学业。

在编辑器中编辑下列命令，进行判断：

```
if ((attendance>=0.90) & (grade>=60))      % 判断出席率和成绩
  pass = 1;                                % 满足条件为通过
else
  fail = 1;                                % 否则为不通过
end
```

3. if-elseif- end 语句

在 else 子句中也可嵌套 if 语句，构成 elseif 结构，elseif 结构实际上实现了多重条件选择，其一般形式为：

if 表达式 1
　　语句体 1;
elseif 表达式 2
　　语句体 2;
else
　　语句体 3;
end

在这种结构中，首先计算表达式 1，如果条件满足执行语句体 1，然后跳出 if 结构，如果不满足表达式 1 的条件，再计算表达式 2，如果表达式 2 的条件满足则执行语句体 2，然后跳出 if 结构，如果前面的表达式都不满足，就执行语句体 3。

根据程序设计的需要可以使用多个 elseif 语句，也可以省略 else 语句。

【例 8-8】　用色彩区分数据点的范围。

在编辑器中编辑下列命令，计算并绘图（图 8-16）：

```
n=100;
x=1:n;
y=randn(1,n);                     % 建立 100 个元素的随机行矢量
hold on
for i=1:n
  if y(i)<-1
    plot(x(i),y(i),'*g')          % 小于-1 的点用绿色的*标出
  elseif y(i)>=-1 & y(i)<=1
    plot(x(i),y(i),'ob')          % 在-1 与 1 之间的点用兰色的 o 标出
  elseif y(i)>=1
    plot(x(i),y(i),'xr')          % 大于 1 的点用红色的 x 标出
  end
end
hold off
```

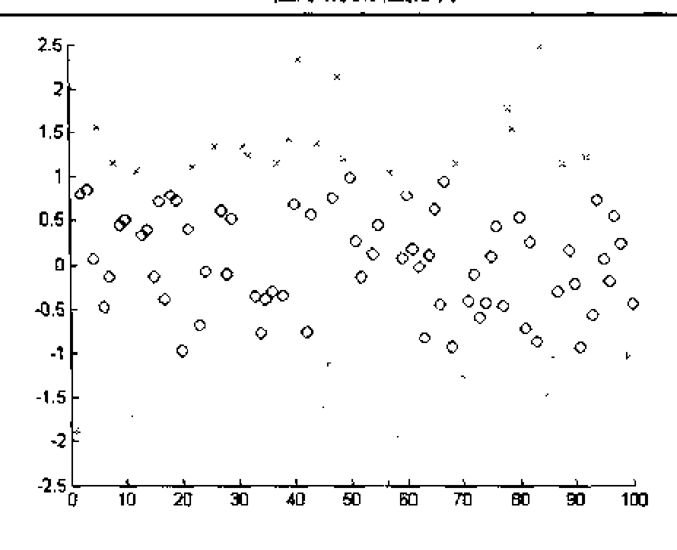


图 8-16 用色彩及标记区分数据点的范围

8.3.3 分支语句

分支语句 switch-case-end 通过对某个表达式的值进行比较，根据比较的结果做不同的选择，以实现程序的分支功能，语句的一般形式为：

switch 表达式（数值或字符串）

case 数值或字符串 1

  语句体 1；

case 数值字或符串 2

  语句体 2；

...

otherwise

  语句体 n；

end

switch 后面表达式的值为数值变量或字符变量，通过这些值与 case 后面的值进行比较，与哪一个 case 的值相同就执行哪一个 case 下面的语句体，如果与所有 case 的值都不相同，则执行 otherwise 下面的语句体。otherwise 语句可以省略，如果省略 otherwise，所有 case 都不满足时跳出分支结构，switch 与 end 必须配对使用。

【例 8-9】 使用 switch-case-end 语句，检查输入数据的值。

在编辑器中编辑下列命令，检查输入数据的值：

```
switch input_num
    case -1
        disp('negative one');              % 当 input_num=-1 时显示
    case 0
        disp('zero');                      % 当 input_num=0 时显示
    case 1
        disp('positive one');              % 当 input_num=1 时显示
    otherwise
         disp('other value');              % 当 input_num 等于其他值时显示
```

```
end
```

与 C 语言不同的是，C 语言中对每个 case 都要比较，即使前面一个已经执行过，因此通常在 case 的语句体后面加一个 break 语句，使程序只执行第一个满足条件的 case。而 MATLAB 中只完成第一个满足条件的 case，不再继续比较，因此不需要 break 语句。

8.3.4　检测语句

检测语句 try-catch 主要用于检测错误并改变流程，其一般形式为：

```
try
    语句体 1
catch
    语句体 2
end
```

检测语句首先执行 try 下面的语句体 1，如果没有错误，执行完成后跳出结构。如果执行语句体 1 时出现错误，则执行 catch 下面的语句体 2。可以使用 lasterr 函数查询最后的错误信息，当查询结果为空字符串时表示语句体 1 成功执行。

【例 8-10】　使用 try-catch 语句，判断错误语句。

在编辑器中编辑下列命令，判断错误语句：

```
n=4;
a=magic(3)                          %设置 3×3 矩阵 a
try
   a_n=a(n,:),                      %取 a 的第 n 行元素
catch
   a_end=a(end,:),                  %如果取 a 的第 n 行出错，则改取 a 的最后一行
end
lasterr                             %显示出错原因
a =
     8     1     6
     3     5     7
     4     9     2
a_end =
     4     9     2
ans =
     Index exceeds matrix dimensions.
```

8.3.5　其他流程控制语句

1. continue 语句

continue 语句用于在 for 循环和 while 循环中跳过某些执行语句。在 for 循环和 while 循环中，如果出现 continue 语句，则跳过循环体中所有剩余的语句，继续下一次循环，在嵌套循环中，continue 控制执行本嵌套中的下一次循环。

在下面的例子中，使用 while 循环统计 magic.m 文件的行数，使用 continue 语句剔除文件中的空行和注释行，即在遇到空行或注释行时，continue 语句跳出当次循环而不计数，最后得到的是文件剔除空行和注释行后的行数。

【例 8-11】 使用 continue 语句，显示文件行数。

在编辑器中编辑下列命令，判断文件行数：

```
fid = fopen('magic.m','r');                 %打开 magic.m 文件
count = 0;                                  %计数器置零
while ~feof(fid)                            %while 循环的判断条件：是否文件末尾
    line = fgetl(fid);                      %从文件中读行
    if isempty(line) | strncmp(line,'%',1)
                                            %判断是否为空行或注释行
        continue                            %如果是空行或注释行继续下一次循环
    end
    count = count + 1;                      %如果不是空行或注释行，计数器加 1
end
    disp(sprintf('%d lines',count));%显示剔除后的行数
```

2. break 语句

break 语句用于终止 for 循环和 while 循环的执行。如果遇到 break 语句，则退出循环体，执行循环体外的下一行语句。在嵌套循环中，break 只存在于最内层的循环中。

下面的例子中，用 while 循环将 fft.m 文件中的内容读入 MATLAB 字符数组中，当遇到第一个空行时，执行 break 语句退出 while 循环，这时结果的字符数组为 fft 函数的帮助信息。

【例 8-12】 break 语句的用法。

在编辑器中编辑下列命令，运行文件：

```
fid=fopen('fft.m','r');             % 打开 fft.m 文件
s='';                               % 创建空的字符数组
while ~feof(fid)                    % while 循环的判断条件：是否文件末尾
    line=fgetl(fid);                % 从文件中读行
    if isempty(line)                % 如果是空行
        break                       % 退出循环
    end
    s=strvcat(s,line);              % 如果不是空行，将该行写入字符数组
end
disp(s)                             % 显示字符数组结果
```

3. return 语句

return 语句用于终止当前的命令序列，并返回到调用的函数或键盘，也用于终止 keyboard 方式。在 MATLAB 中，被调用的函数运行结束后会自动返回到调用函数，使用 return 语句时将 return 插入被调用函数的某一位置，根据某种条件迫使被调用函数提前结束并返回调用函数。

在计算行列式的函数中，可以用 return 语句处理如空矩阵之类的特殊情况。

【例 8-13】 return 语句的用法。

在编辑器中编辑下列函数，运行文件：

```
function d = det(A)                    % 定义函数
% det(A) 为计算矩阵 A 的行列式的函数
if isempty(A)                          % 如果矩阵是空的
    d = 1;
    return                             % 返回调用函数
else
    ...                                % 计算过程这里省略
end
```

8.4　程序的调试与优化

在程序设计中不可避免地会出现各种各样的错误，一般来说可以归纳为两种：语法错误和运行错误。

语法错误通常是在程序输入时产生的，如函数名拼写错误、括号不匹配等问题。由于这些错误的存在，程序不能完成全部运行过程，会在发现错误时停止运行，并在命令窗口显示错误信息，因此这类错误在运行的过程中就能够被发现，并可以通过对程序的调试较容易地得到修改。

运行错误通常是在算法上发生了错误，例如错误地修改某个变量或完成一个不正确的计算，当产生这些意想不到的错误结果时，就出现了运行错误。

另外还有一些错误可能是由于解题思路不正确或对问题的理解不准确而引起的，通常在运行时不会有错误信息输出，只有发现计算结果有较大的偏差或不符合设计要求时，才能根据结果的差异进行分析和判断，这可能是一个比较复杂的过程。在这里只介绍语法错误和运行错误的调试方法。通常对于简单的程序可以用直接调试的方法进行修改，对于一些比较复杂程序的调试则需要借助于程序调试器（Debugger）。

8.4.1　程序的直接调试法

对于简单的程序，可以进行直接调试与修改，直接调试法包括以下几个方面：

(1) 如果在错误信息中指出了出错的行数，可先根据错误信息检查该语句是否存在语法错误或运算中变量尺寸不一致等情况。

(2) 检查所调用的函数或命令的拼写是否正确，括号（包括方括号和圆括号）是否配对，各种流程控制语句是否匹配（如 for 与 end、while 与 end、switch 与 end 等）。

(3) 检查所调用的函数或要载入的数据文件是否在当前目录或搜索路径上。

(4) 将重点怀疑的命令行后的分号删除或改成逗号，使得计算结果能够实时地显示在屏幕上，根据显示的结果判断问题的所在。

(5) 利用 echo 命令，可以在运行时将文件的内容显示在屏幕上。echo on 用于显示命令文件的执行过程，但不显示被调用函数文件的内容，如果希望检查函数文件中的内容，用 echo Funname on 显示文件名为 Funname 的函数文件的执行过程。一般情况下，为了能够观察清楚，建议不要同时使用 echo on 和 echo Funname on，或同时显示多个函数文件的内容，分别检查会更容易查找错误的位置。echo off 用于关闭命令文件的执行过程显示，echo Funname off

用于关闭函数文件的执行过程显示。

(6) 在程序文件的适当位置添加 keyboard 命令，当运行中的程序遇到 keyboard 命令时会暂时停止运行，将控制权交给键盘，这时命令窗口的提示符变为 K>>，可以在命令窗口键入变量名，以检查各变量的内容是否正确，也可键入命令修改变量。检查或修改完成后，键入 return 命令，控制权交还给系统，程序将继续运行。

(7) 如果怀疑某个函数文件有问题，可以在该函数文件的函数定义行前加%，使之变为可以观察中间变量的命令文件（因为命令文件中的变量存储在工作空间中，可以在工作空间浏览器和数组编辑器观察和修改）。

如果函数文件规模较大，文件嵌套复杂，或调用较多的函数时，可以借助于专门的调试工具——程序调试器（Debugger）。

8.4.2　调试器的使用

调试器位于编辑/调试器窗口中，关于该窗口的功能和使用参见 1.4。使用调试器调试程序的一般步骤如下：

(1) 在 Debug 菜单中选择 Run 开始运行。

(2) 出现错误时，运行自动停止，根据错误信息找到相应的程序位置。MATLAB 6.1 版本提供了一个方便的功能，当在命令窗口出现了带有下划线的错误信息时，用鼠标点击带下划线部分，光标将自动找到程序中的错误行，见图 8.17 和图 8.18。

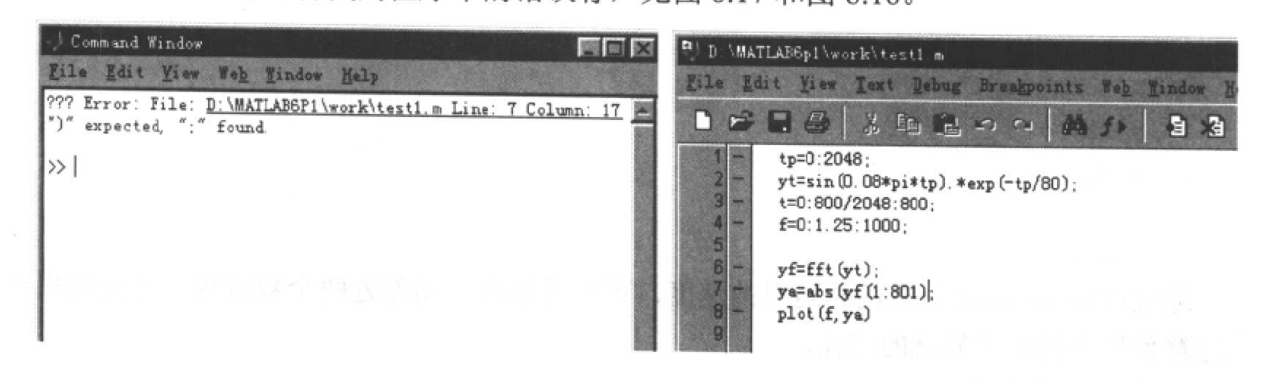


图 8.17　显示在命令窗口中的错误信息　　　　图 8.18　光标指示在错误程序行的末尾

根据命令窗口中的错误信息并检查光标指示的命令行，可以很容易地查出错误的原因。

(3) 在可能出现错误的程序行位置添加断点。如果错误在被调用的函数中，断点应设置在被调用的函数前。

(4) 重新运行后程序会自动暂停在断点处，将鼠标指向感兴趣的变量名，可以看到变量的内容，此时相当于处于 keyboad 状态，也可以从命令窗口检查变量的的大小、内容等信息。

(5) 如果希望进入被调用的函数内部进行观察，选择 Step in 图标，直至查出错误，进行修改。

(6) 重新运行，检查是否有新的错误。

设置断点是高级语言中程序调试的重要手段之一，断点是在程序中特定位置设定的中断点，当程序运行至某一断点处时会暂停运行，此时可通过检查相关变量的内容等方法确定程序的运行是否正确。在断点处一般可以控制程序按程序行逐行向后继续运行，也可以控制程

序继续运行到指定的程序行。根据需要，可以在程序中设置一个或多个断点。

8.4.3　程序设计的优化

尽管 MATLAB 已将大多数的运算和操作集成到功能强大的函数中，但通过对 MATLAB 程序的优化设计可以进一步提高 MATLAB 程序的运行效率。

1. 以矩阵运算代替循环运算

循环运算是 MATLAB 语言的一个弱点，在程序设计时，应当尽可能避免循环运算。由于矩阵运算是 MATLAB 的核心，因此，在 MATLAB 编程过程中应当强调对矩阵的整体运算，减少和避免对矩阵元素的操作，尽可能将循环运算转换为矩阵运算。

【例 8-14】　循环运算和矩阵运算的比较。

分别进行 10 000 个元素的循环运算和矩阵运算，比较同一问题求解所用的时间：

```
>> t=cputime;
>> for i=1:10000
       x(i)=0.1*pi*i;
       y(i)=sin(i);
   end
>> e=cputime-t
e =
   7.4700
>> t=cputime;
>> x=1:0.1*pi:1000*pi;
>> y=sin(x);
>> e=cputime-t
e =
    0.0600
```

通过函数 cputime 分析两种运算的速度，可以看到两者相差近两个数量级，可见使用矩阵运算使程序得到了显著的优化。

2. 数据的预定义

虽然在 MATLAB 中没有规定变量使用时必须预先定义，但是对于未定义的变量，如果操作过程中出现越界赋值等问题，系统将不得不对变量进行扩充，这样的操作大大降低了程序运行的效率。所以对于可能出现变量维数不断扩大的问题时，应当预先估计变量可能出现的最大维数，进行预定义。

【例 8-15】　变量是否预定义的比较。

进行 10 000 个元素的循环运算，比较预定义矩阵尺寸和未进行预定义时所用的时间：

```
>> x=zeros(10000,1);
>> y=zeros(10000,1);
>> t=cputime;
>> for i=1:10000
       x(i)=0.1*pi*i;
       y(i)=sin(i);
```

```
    end
>> e=cputime-t
e =
    0.5500
```

通过函数 cputime 分析是否预定义矩阵时的运算速度，可以看到两者相差近一个数量级，可见预定义矩阵时可以使程序得到了显著的优化。

3. 内存的管理

对内存的合理操作及管理也能提高程序运行的效率。

MATLAB 提供了一系列用于内存管理的函数，这些函数见表 8-4。

表 8-4　MATLAB 语言中的内存管理函数

函数名	说明
clear	从内存中清除所有变量及函数
pack	重新分配内存
quit	退出 MATLAB 环境，释放所有内存
save	把指定的变量存储至磁盘
load	从磁盘中调出指定变量

在上述命令中，pack 函数最为重要，该函数将内存中所有 MATLAB 使用的变量暂存入磁盘，然后再用内存中连续的空间存贮这些变量，由于要与磁盘之间进行数据交换，所以该命令的执行速度较慢，一般不在函数内部使用。但是，在进行计算的过程中，若出现 out of memory 的错误，通过该命令重新分配内存，可以在一定程度上解决问题。

此外，应当指出的是，MATLAB 本身不具备管理系统资源的能力，所以，在进行较大规模的计算时，应尽可能关闭一切不必要的应用程序，以节省资源。

8.4.4 M 文件调用记录

MATLAB 6 提供了记录 M 文件调用过程的功能。通过记录 M 文件调用过程，可以分析执行过程中各函数的耗时情况，依此可以了解文件执行过程中的瓶颈问题。通过 M 文件调用记录可以避免许多程序设计中不必要的冗余以及时间耗费。

1. profile 函数

实现 M 文件调用记录的函数为 profile，其调用格式为：

- profile on　开始记录 M 文件的调用，并清除以前的记录。
- profile on –detail level　按函数指定的 level 开始记录 M 文件的调用，并清除以前的记录。
- profile on –history　记录确定序列的函数调用。
- profile off　中断 M 文件调用记录。
- profile resume　重新开始 M 文件调用记录，并保存原来的记录。
- profile clear　清除 M 文件调用记录。
- profile report　中断 M 文件调用记录，并将记录以 html 格式输出。

• profile report basename　中断 M 文件调用记录，并将记录保存至当前目录下 basename 的文件中。

• profile plot　中断 M 文件调用记录，并将结果以条状图绘图。

• s = profile('status')　显示当前调用的状态。

• stats = profile('info')　中断并返回记录结果。

2. 调用记录结果的显示

下面以求解微分方程为例，介绍调用记录的使用方法。

【例 8-16】　调用记录的应用实例。

在命令窗口执行下列命令：

```
>> profile on -detail builtin -history
>> [t,y]=ode23(@vdp1,[0 20],[2 0]);
>> profile report ode_report
>> profile plot
```

通过 profile report 函数可以将记录输出到 ode_report.html 文件中，如图 8-19 所示。该调用记录的结果由 3 部分页面组成，分别为调用记录综述、函数调用明细及函数调用历史。

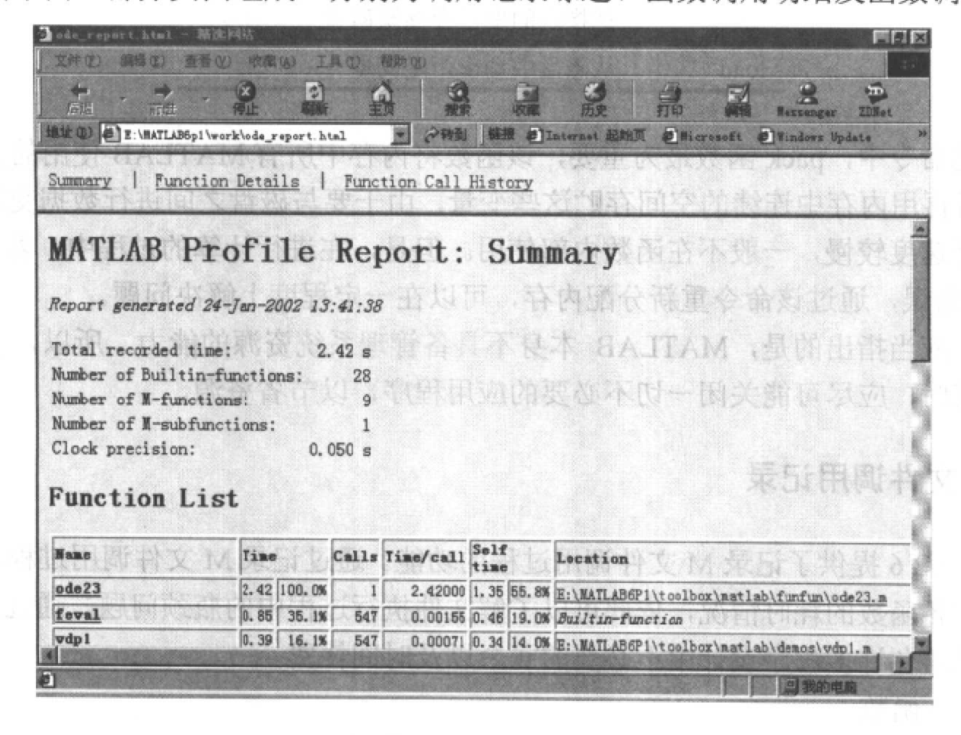


图 8-19　调用记录综述

调用记录综述中显示包括被调用函数（包括内置函数、函数以及子函数）列表、统计时间精度、时间统计以及使用时间统计等，其中调用函数列表包含以对象函数的形式调用的所有函数，统计时间精度给出了优化过程计时的最小单位；时间统计中则给出函数列表内每个函数总的调用时间，即包含函数内部的子函数所耗用的时间；使用时间统计则给出了每个函数执行过程中在本函数体内的时间，即不包括子函数调用时耗用的时间。

在函数调用明细中，给出每个函数的细节，如图 8-20 所示。在该页面中显示调用的所有函数的细节，包括被调用次数、调用函数以及其子函数，该函数的时间统计以及使用时间统计，各语句的执行情况等。

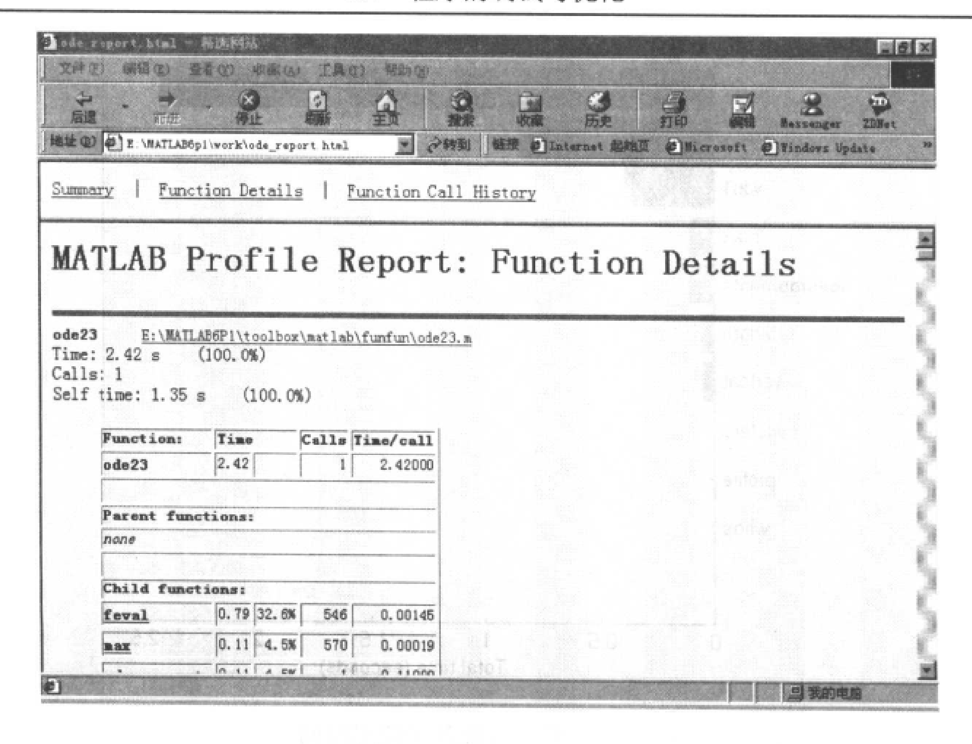


图 8-20 调用记录明细

图 8-21 为函数调用历史界面，给出所调用的内置函数、子函数的列表等。

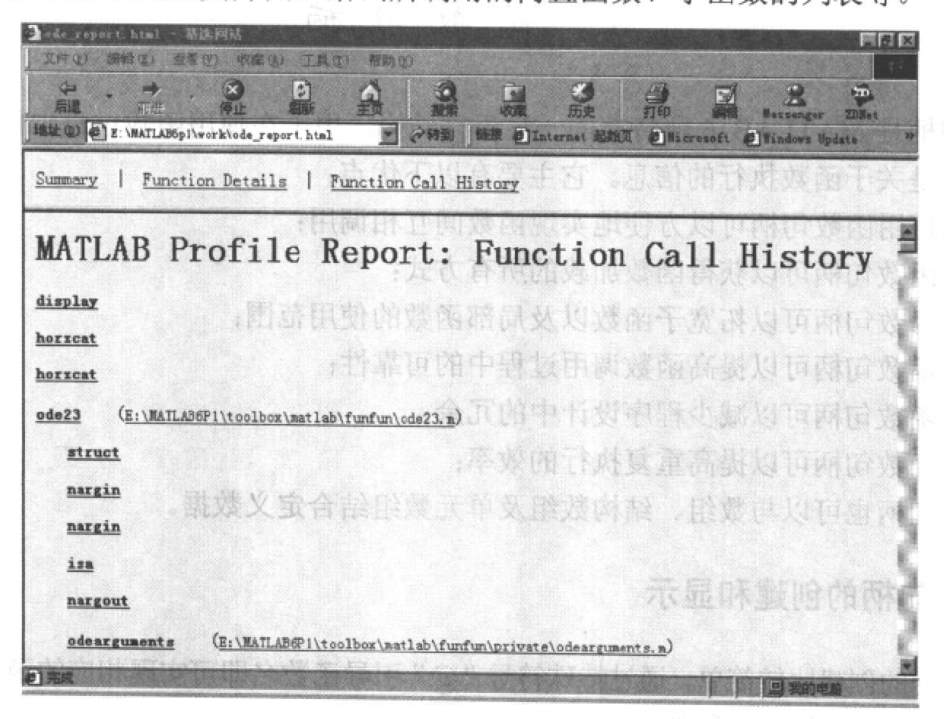


图 8-21 调用记录历史

另外，通过 plofile plot 还可以将调用历史记录显示为条状图输出，如图 8-22 所示。在该图中显示主要调用函数及其调用时间，并以百分比条状图显示。

通过对调用记录结果的分析，可以掌握 M 文件在执行过程中的信息，对于进一步进行程序的优化设计具有很大的意义。

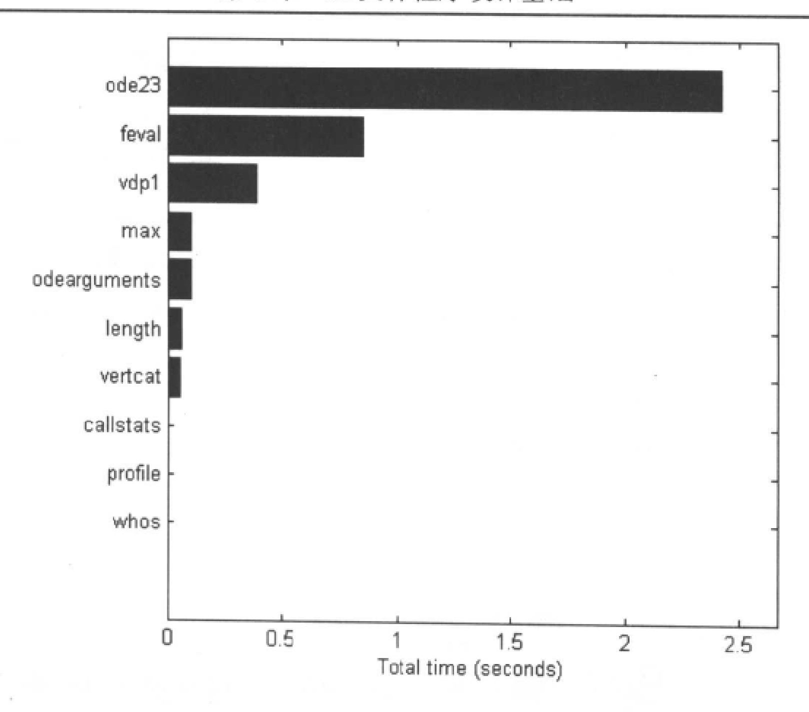


图 8-22　调用函数及其调用时间

8.5　函 数 句 柄

函数句柄是 MATLAB 6 所特有的一种语言结构，用于在使用函数过程中保存函数相关信息，尤其是关于函数执行的信息。它主要有以下优点：

- 通过使用函数句柄可以方便地实现函数间互相调用；
- 通过函数句柄可以获得函数加载的所有方式；
- 通过函数句柄可以拓宽子函数以及局部函数的使用范围；
- 使用函数句柄可以提高函数调用过程中的可靠性；
- 通过函数句柄可以减少程序设计中的冗余；
- 通过函数句柄可以提高重复执行的效率；
- 函数句柄也可以与数组、结构数组及单元数组结合定义数据。

8.5.1 函数句柄的创建和显示

函数句柄的创建比较简单，通过特殊符号“@”引导函数名即可实现相应的函数句柄定义。

【例 8-17】　创建 plot 函数的函数句柄。

在命令窗口输入命令，创建函数句柄：

```
>> h01=@plot
h01 =
    @plot
```

定义函数句柄时只需在符号“@”后添加相应的函数名（注意，不能将函数的路径也填写在内，否则将会出错）。

函数句柄的内容可以通过函数 functions 来显示，使用该函数将返回函数句柄所对应的函数名、类型、文件类型以及加载方式等。其中函数类型如表 8-5 所示，函数的文件类型是指该函数句柄的对应函数是否为 MATLAB 内部函数，而函数的加载方式属性只有当函数类型为 overloaded 时才存在。

表 8-5 MATLAB 的函数类型

函数类型	说明
simple	未加载的 MATLAB 内部函数、M 文件，或只有在执行过程中才能用 type 函数显示内容的函数
subfunction	MATLAB 子函数
private	MATLAB 局部函数
constructor	MATLAB 类的创建函数
overloaded	加载的 MATLAB 内部函数或 M 文件

【例 8-18】 显示函数句柄的内容。

在命令窗口使用 functions 函数显示函数句柄的信息：

```
>> functions(h01)
ans =
    function: 'plot'
        type: 'overloaded'
        file: 'MATLAB built-in function'
    methods: [1x1 struct]
```

从工作空间可以看出函数句柄的内容用一个结构数组表示，同时该函数句柄的加载方式也是一个结构数组（图 8-23）。

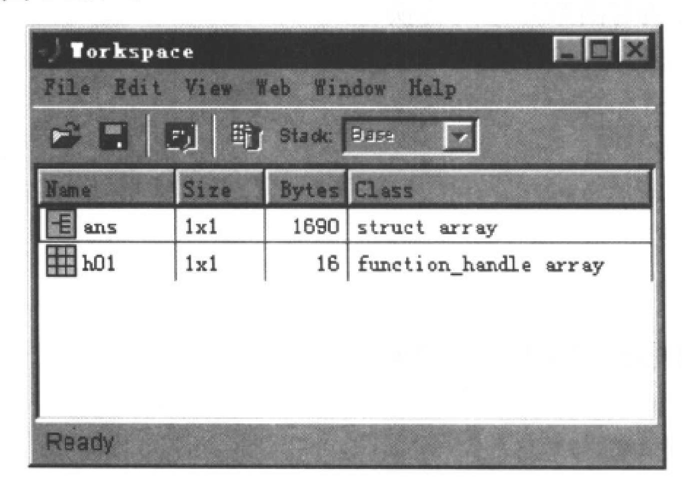


图 8-23 工作空间中的函数句柄

8.5.2 函数句柄的调用和操作

使用 feval 函数可以实现函数句柄的调用，其调用格式为：

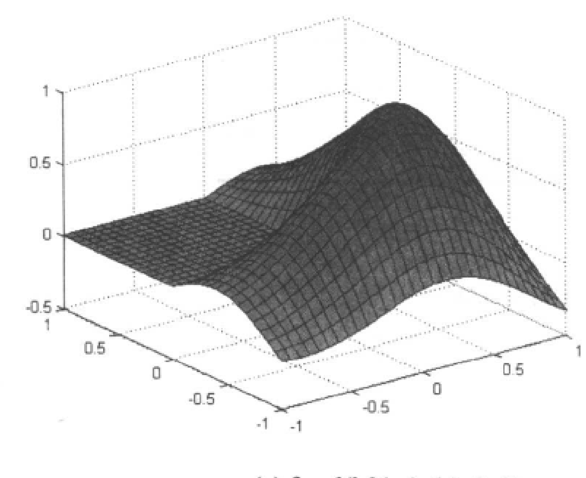
- [*y*1, *y*2, ...] = feval(*fhandle*, *x*1, ..., *xn*) 执行函数句柄 *fhandle*，使用的参数为 *x*1, ..., *xn*。

这种调用相当于执行以参数列表为输入变量的函数句柄所对应的函数。

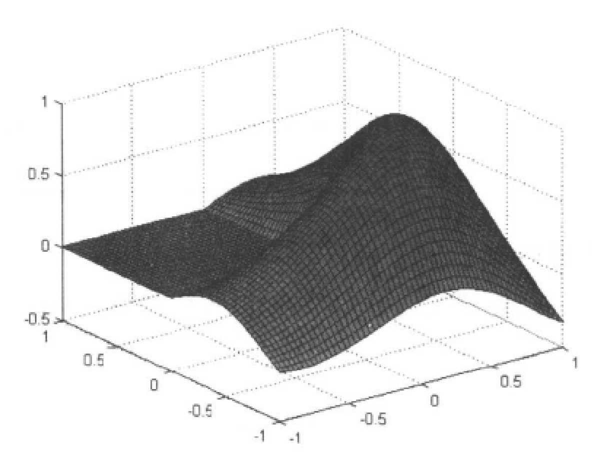
【例 8-19】　创建 membrane 函数的函数句柄，并运行该函数。

在命令窗口创建 membrane 函数的函数句柄的信息，使用不同参数调用所创建的函数句柄（图 8-24）：

```
>> h01=@membrane
h01 =
    @membrane
>> feval(h01,1,15,9,2)
>> feval(h01,1,30,9,2)                    % 加密网格绘图
```



(a) feval(h01, 1, 15, 9, 2)



(b) feval(h01, 1, 30, 9, 2)

图 8-24　使用不同参数调用函数句柄

函数句柄与函数名字符串之间可以进行转换，转换函数为 func2str 和 str2func。

【例 8-20】　函数句柄与函数名之间转换。

在命令窗口使用 str2func 和 func2str 函数进行转换：

```
>> h01=str2func('plot')
h01 =
    @plot
>> func2str(h01)
ans =
    plot
```

通过函数 isa 可以判断变量是否为函数句柄。

```
>> isa(h01,'function_handle')
ans =
     1
```

通过函数 isequal 可以判断两函数句柄是否相同。

```
>> isequal(h01,@plot)
ans =
     1
```

同其他的 MATLAB 数据相同，通过函数 save 也可以将函数句柄保存为 MATLAB 数据文件，而使用函数 load 则可以调入函数句柄。

习 题 8

8.1 命令文件与函数文件的主要区别是什么？

8.2 如何定义全局变量？

8.3 如果 x 是一个结构数组，如何观察其中的内容？

8.4 if 语句有几种表现形式？

8.5 通过编程说明 break 语句和 return 语句的用法。

8.6 有一周期为 4π 的正弦波上叠加了方差为 0.1 的正态分布的随机噪声的信号，用循环结构编制一个 3 点线性滑动平均的程序。（提示：①用 0.1*randn(1, n)产生方差为 0.1 的正态分布的随机噪声；②3 点线性滑动平均就是依次取每 3 个相邻数的平均值作为新的数据，如 $x1(2)=(x(1)+x(2)+x(3))/3$，$x1(3)=(x(2)+x(3)+x(4))/3$，……）

8.7 编制一个解数论问题的函数文件：取任意整数，若是偶数，则除以 2，否则乘 3 加 1，重复此过程，直到整数变为 1。

8.8 有一组学生的考试成绩（见表）。根据规定，成绩在 100 分时为满分，成绩在 90~99 之间为优秀，成绩在 80~89 分之间为良好，成绩在 60~79 分之间为及格，成绩在 60 分以下时为不及格。编制一个根据成绩划分等级的程序。

学生姓名	王	张	刘	李	陈	杨	于	黄	郭	赵
成 绩	72	83	56	94	100	88	96	68	54	65

8.9 编写一段程序，能够把输入的摄氏温度转化成华氏温度，也能把华氏温度转换成摄氏温度。

8.10 keyboard 命令的作用是什么？退出 keyboard 状态的命令是什么？当在函数中使用 keyboard 命令时，是否可以在工作空间浏览器中观察和修改函数中的局部变量？编制一个函数文件，在其中设置 keyboard，观察变量情况。

第9章　Simulink 基础

Simulink 是实现动态系统建模和仿真的集成环境，其主要的功能是对动态系统进行仿真和分析，预先模拟实际系统的特性和响应，根据设计及使用的要求，对系统进行修改和优化，以提高系统的性能，实现高效开发系统的目标。

9.1　Simulink 初步

9.1.1　Simulink 简介

使用传统的方法实现系统的模拟和仿真，需要将系统的模块函数转化为微分或差分方程，再用某种程序设计语言（如 C 语言等）编程进行仿真运算。Simulink 提供了图形化的用户界面，用户只需点击鼠标，就可以轻易地完成模型的创建、调试和仿真工作，这样就大大降低了仿真的难度，用户不需为了完成仿真工作而去学习某种程序设计语言。

Simulink 提供了大量的系统模块，包括了信号、运算、显示和系统等多方面的功能，可以创建各种类型的仿真系统，实现丰富的仿真功能。用户也可以定义自己的模块，进一步扩展模型的范围和功能，以满足不同的需求。

为了创建大型系统，Simulink 提供了系统分层排列的功能，类似于系统的设计。在 Simulink 中可以将系统分为高级到低级的几个层次，每层又可以细分为几个部分，每层系统构建完成后，将各层连接起来构成了一个完整的系统。

模型创建完成之后，可以启动系统的仿真功能分析系统的动态特性。Simulink 内置的分析工具包括各种仿真算法、系统线性化、寻求平衡点等，仿真结果可以以图形的方式显示在示波器窗口，以便于用户观察系统的输出结果。Simulink 的输出结果可以以变量的形式保存起来，并输入到 MATLAB 中以完成进一步的分析。

Simulink 可以仿真线性或非线性系统，并能够创建连续时间、离散时间或二者混合的系统。Simulink 还支持多采样频率系统，即不同的系统能够以不同的采样频率进行组合，可以仿真较大较复杂的系统。

模型的创建与定义、模型的分析以及修正是使用 Simulink 的三大步骤，图 9-1 显示了典型的 Simulink 工作框图。

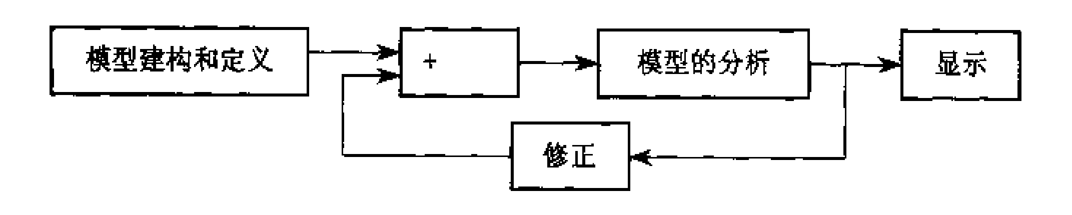


图 9-1　Simulink 操作框图

9.1.2　运行 Simulink

在 MATLAB 命令窗口下，点击工具栏中的图标（或键入命令 simulink），就会弹出一个名为“Simulink Library Browser”的浏览器窗口（图 9-2），该窗口以树状列表的形式列出了当前 MATLAB 系统中已安装了的所有 Simulink 的模块，这些模块包含 Simulink 模块库中的各种模块及其他 Toolbox 和 Blockset 中的模块。展开树状列表，用鼠标点击所需的模块项，列表窗口的上方会显示所选模块的信息，也可以在浏览器窗口右上角的输入栏中直接键入模块名并按 Find 按钮进行查询。如果输入一个不存在的模块，将弹出对话框给出提示。

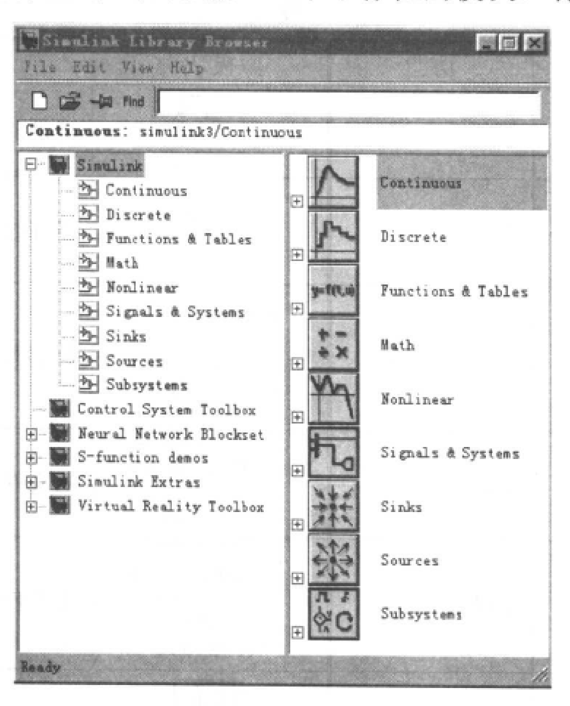


图 9-2　Simulink 库浏览器

在创建新模型时，先在浏览器上方的工具栏中选择“建立新模型”的图标，则弹出一个名为“Untitled”（无标题）的空白窗口，所有的控制模块都创建在这个窗口中。

9.2　Simulink 的基本模块

Simulink 的模块库提供了大量模块。

在 Simulink 的浏览器窗口左侧的 Simulink 项上单击鼠标右键，单击弹出菜单“Open the ‘Simulink’ Labrary”，将打开一个独立的 Simulink 模块库窗口（图 9-3），下面对各子模型库作介绍。

9.2.1　信号源模块库（Source）

信号源模块库包括 17 种信号源模块，为仿真系统提供了不同的信号输入方法。信号源模块的功能参见表 9-1。

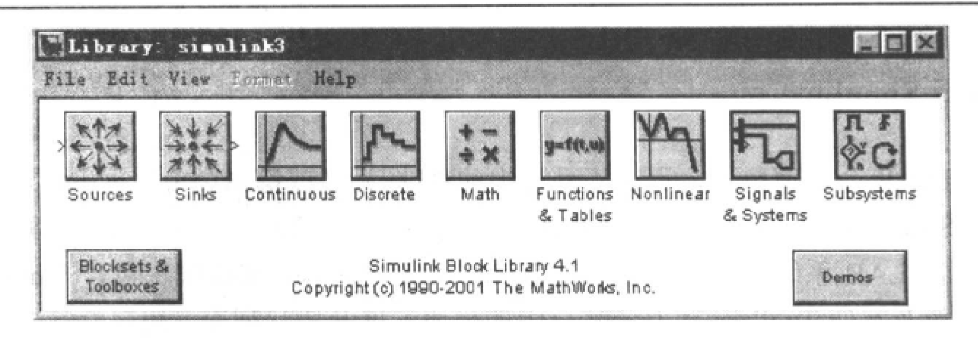


图 9-3　Simulink 模块库窗口

表 9-1　信号源模块的功能

模块	功能	模块	功能
In1	创建输入端	Ground	接地
Constant	常数	Clock	当前时间
Signal Generator	信号发生器	Digital Clock	数字时钟
Ramp	斜波	From File	从文件读数据
Sine Wave	正弦波	From Workspace	从工作空间读数据
Step	阶跃信号	Random Number	随机信号
Repeating Sequence	重复序列	Uniform Random Number	均匀随机信号
Pulse Generator	脉冲发生器	Band-Limited White Noise	带限白噪声
Chirp Signal	快速正弦扫描		

9.2.2　输出模块库（Sinks）

输出模块库提供了各种输出模块，包括图形的显示和数据的存储等。输出模块的功能参见表 9-2。

表 9-2 输出模块的功能

模块	功能	模块	功能
Scope	示波器	untitled.mat To File	输出到文件
Floating Scope	可选示波器	simout To Workspace	输出到工作空间
XY Graph	XY关系图	Terminator	通用终端
1 Out1	创建输出端	STOP Stop Simulation	输入不为 0 时停止仿真
0 Display	实时数值显示		

9.2.3 连续系统模块库（Continuous）

连续系统模块库提供了连续系统运算的模块。连续系统模块的功能参见表 9-3。

表 9-3 连续系统模块的功能

模块	功能	模块	功能
1/s Integrator	积分	(s-1)/s(s+1) Zero-Pole	零极点
du/dt Derivative	微分	Memory	延时输出
x' = Ax+Bu y = Cx+Du State-Space	状态方程	Transport Delay	传输延时
1/(s+1) Transfer Fcn	传递函数	Variable Transport Delay	可变传输延时

9.2.4 离散系统模块库（Discrete）

离散系统模块库提供了滤波器、传递函数等离散系统模块。离散系统模块的功能参见表 9-4。

表 9-4　离散系统模块的功能

模块	功能	模块	功能
Zero-Order Hold	零阶保持器	$\frac{1}{1+0.5z^{-1}}$ Discrete Filter	离散滤波器
$\frac{1}{z}$ Unit Delay	单位延迟采样保持	$\frac{1}{z+0.5}$ Discrete Transfer Fcn	离散传递函数
$\frac{T}{z-1}$ Discrete-Time Integrator	离散时间积分	$\frac{(z-1)}{z(z-0.5)}$ Discrete Zero-Pole	离散零极点
y(n)=Cx(n)+Du(n) x(n+1)=Ax(n)+Bu(n) Discrete State-Space	离散状态方程	First-Order Hold	一阶保持器

9.2.5　数学运算模块库（Math）

数学运算模块提供了各种用于数学运算的模块，包括数学运算、关系运算、逻辑运算和复数运算等。数学运算模块的功能参见表 9-5。

表 9-5　数学运算模块的功能

模块	功能	模块	功能
Sum	求和	floor Rounding Function	取整函数
× Product	积或商	Combinatorial Logic	逻辑真值表
• Dot Product	点积	AND Logical Operator	逻辑算子
1 Gain	常数增益	bitwise AND 'FFFF' Bitwise Logical Operator	位逻辑算子
1 Slider Gain	可变增益	<= Relational Operator	关系算子

续表

Matrix Gain	矩阵增益	Complex to Magnitude-Angle	复数的模和辐角
Math Function	数学运算函数	Magnitude-Angle to Complex	模和辐角合成复数
Trigonometric Function	三角函数	Complex to Real-Imag	复数的实部和虚部
MinMax	求最大值	Real-Imag to Complex	实部和虚部合成复数
Abs	求绝对值	Algebraic Constraint	强迫输入信号为零
Sign	符号函数		

9.2.6 函数和表模块库（Functions & Tables）

函数和表模块提供了插值、多项式等模块。函数和表模块的功能参见表 9-6。

表 9-6 函数和表模块的功能

模块	功能	模块	功能
Look-Up Table	线性插值查表	Fcn	C 语言形式的表达式
Look-Up Table (2-D)	二维线性插值查表	MATLAB Fcn	MATLAB 形式的表达式
Look-Up Table (n-D)	N 维线性插值查表	S-Function	调用 S–函数

续表

PreLook-Up Index Search	预查下标	Polynomial	多项式
Interpolation (n-D) using PreLook-Up	N 维插值	S-Function Builder	用 C 代码创建 S-函数
Direct Look-Up Table (n-D)	直接查表		

9.2.7　非线性系统模块库（Nonlinear）

非线性系统模块库提供了开关、继电器等模块。非线性系统模块的功能参见表 9-7。

表 9-7　非线性系统模块的功能

模块	功能	模块	功能
Rate Limiter	速率限制器	Relay	继电器
Saturation	饱和元件	Switch	开关
Quantizer	量化元件	Manual Switch	手动开关
Backlash	间隙元件	Multiport Switch	多选开关
Dead Zone	死区元件	Coulomb & Viscous Friction	库仑和粘性摩擦

9.2.8　信号与系统模块库（Signal & Systems）

信号与系统模块库提供了 20 多种用于信号处理的模块。
信号与系统模块的功能参见表 9-8。

表 9-8 信号与系统模块的功能

模块	功能	模块	功能
Bus Creator	创建信号总线	Data Store Memory	为存储器定义内存
Bus Selector	从信号总线中选择信号	Data Store Write	向存储器写数据
Mux	多路复合器	Function-Call Generator	函数调用发生器
Demux	多路分离器	Reshape	改变信号尺寸
Selector	选择输入信号	Data Type Conversion	数据类型转换
Assignment	赋值	Hit Crossing	检测零交叉点
Matrix Concatenation	矩阵串联	IC	信号的初始值
Merge	信号合并	Width	信号的宽度
From	从 GOTO 模块接收信号	Model Info	显示模型信息
Goto Tag Visibility	定义 GOTO 模块的范围	Signal Specification	检查信号参数
Goto	把信号送到 From 模块	Probe	探测连线
Data Store Read	从存储器读数据		

9.3　Simulink 建模

9.3.1　模块的创建及操作

1. 创建模块

在浏览器列表中点击需要的模块，按住鼠标左键并拖曳至模型窗口即可。双击模块，弹出该模块的参数设置对话框，在对话框中修改相应的参数。每个模块的下方都有一个名称，双击名称处，使之处于文本输入状态，即可改变该模块的名称（图 9-4）。

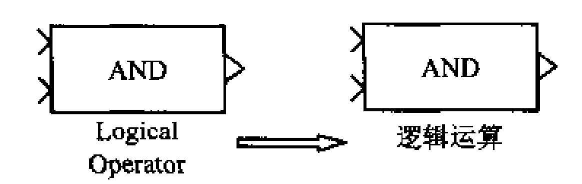


图 9-4　改变模块名称

观察已建立的模块，在模块的左右两侧，分别有不同数量的箭头，左侧向内的箭头为输入端口，用于连接前一级模块，右侧向外的箭头为输出端口，用于连接下一级模块，不同的模块有不同数量的输入和输出端口。

2. 模块操作

(1) 模块的选择。点击要选择的模块，该模块的四周就会出现标记点，表明该模块被选中，下面的各种操作都将作用于选定的模块。如果希望同时选定多个模块，可在窗口内适当位置处按住鼠标左键拖曳出一个框，包含住所要选择的模块，被选定的模块的四周都会显示出控制点标记（图 9-5）。

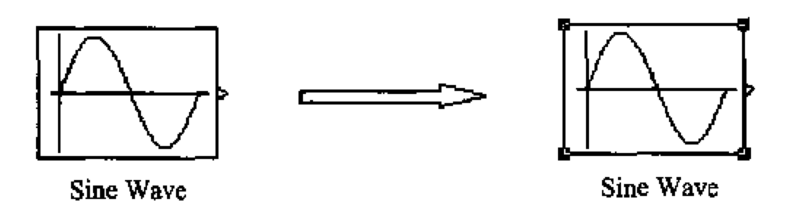


图 9-5　模块的选中

(2) 移动模块。选定要移动的模块，按住鼠标左键并拖动至适当的位置。

(3) 模块的缩放。选定要缩放的模块，在四个标记点中的任意一处，按住鼠标左键并拖动至适当大小即可（图 9-6）。

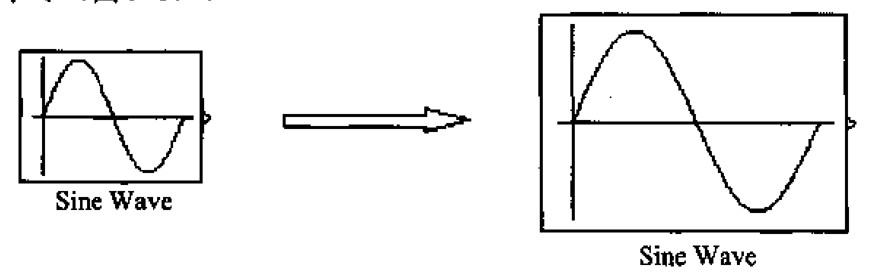


图 9-6　模块的缩放

(4) 复制模块。下列方法均可用于同一模型窗口中模块的复制。

① 选定模块，按下鼠标右键并拖动至适当的位置；

② 选定模块，在工具栏中选择“Copy”与“Paste”按钮；

③ 在选定的模块上点击鼠标右键，在弹出的菜单中选择 Copy 与 Paste 选项；

④ 按住 Ctrl 键，按下鼠标左键，将选定的模块拖动至适当的位置。

在不同的模型窗口中复制模块，除了上述方法均可使用外，还有一种简单的方法，即在选定的模块上按下鼠标左键，将模块拖动至新的窗口中。

(5) 模块的旋转与翻转

旋转　将鼠标指向要旋转的模块并按下鼠标右键，在弹出的菜单中选择 Format 栏中的 Rotate Block 项，模块将顺时针旋转 90°，每选择一次，模块都将顺时针旋转 90°。

翻转　将鼠标指针指向要翻转的模块并按下鼠标右键，在弹出的菜单中选择 Format 栏中的 Flip Block 项，模块将翻转 180°，每选择一次，模块都将翻转 180°（图 9-7）。

上述功能也可以在菜单栏的 Format 菜单中实现。

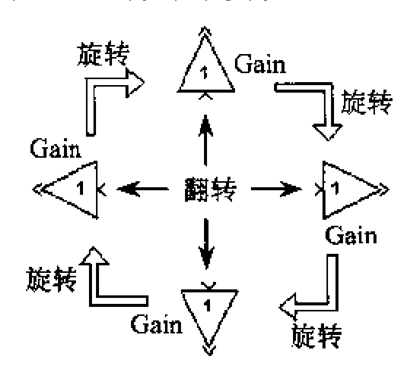


图 9-7　模块的旋转与翻转

(6) 模块的连接。模块之间的连接非常简单，只要将鼠标指针移到模块的输出（或输入）端，指针就会变成十字形状，按住鼠标左键，拖动至另一模块的输入（或输出）端，即可完成连接（图 9-8）。

如果要删除连线，可点击要删除的线，此时线上出现标记点，表示已被选中，然后按工具栏中的剪切按钮即可删除。

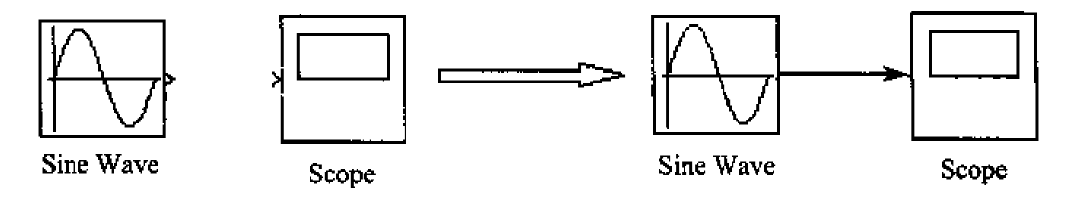


图 9-8　模块的连接

(7) 连接分支线。在已有的连线上的某一个位置按住鼠标右键，可以拖出一条分支线，也可以同时按住 Ctrl 键和鼠标左键并拖动鼠标（图 9-9）。

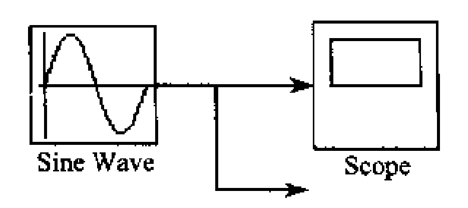


图 9-9　分支线

(8) 改变连线的形状。在连线上按住鼠标左键拖动，可以改变折线的平行位置。选定连线，拖动连线上的标记点，可以改变折线的水平和垂直位置。在按住 Shift 键的同时进行上面的操作，可以拉出斜线（图 9-10）。

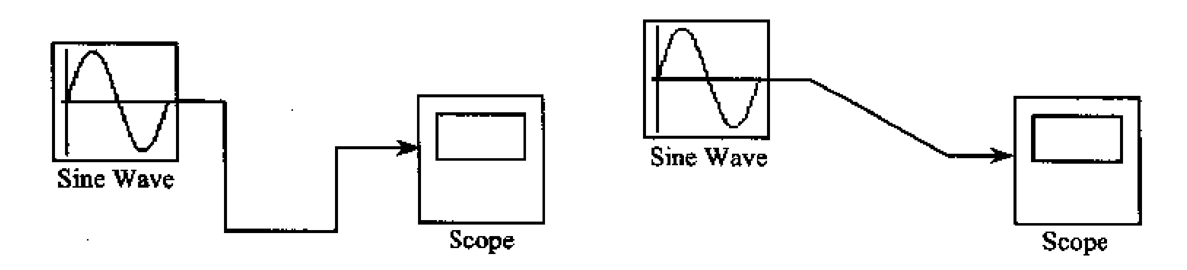


图 9-10　改变连线形状

(9) 连线的标识。在连线的上方或下方双击鼠标左键，会出现一个文本输入框，可输入连线的标识，事实上可以在模型窗口中的任意地方输入文字说明（图 9-11）。

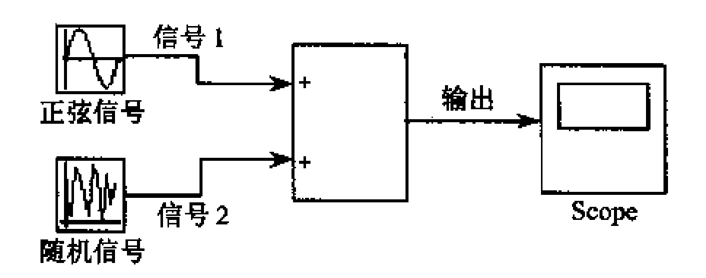
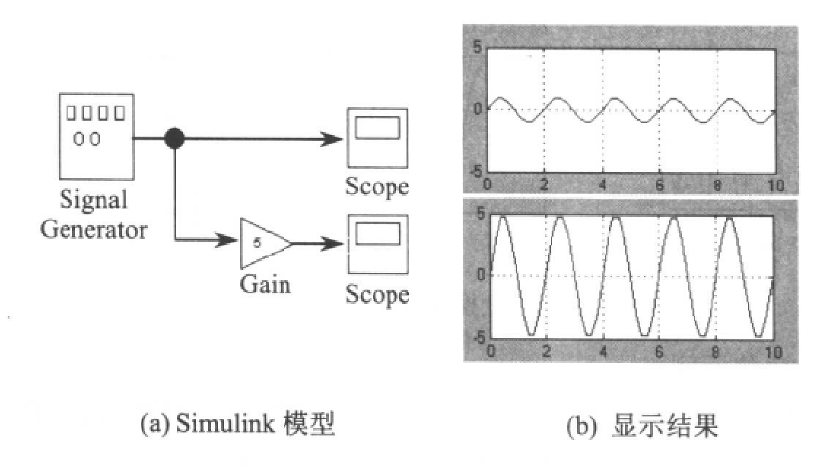


图 9-11　文字说明

不仅可以在连线附近加注标识，还可以在模型窗口的任意位置输入如标题、说明等文本内容，并且可以改变字体的大小与类型。

3. 简单模型

图 9-12 是一个最简单的模型，信号发生器发生幅值为 1、频率为 0.2 Hz 的正弦波信号，信号分为两路，一路直接送到示波器显示，另一路放大 5 倍后送到另一示波器中显示，观察显示结果。



(a) Simulink 模型　　(b) 显示结果

图 9-12　一个简单模型及运行结果

9.3.2　模型的修饰

在模型窗口的 Format 菜单中提供了多种用于修饰模型的选项。

1. 模块加阴影

Format 菜单中的 Show drop shadow 菜单项用于给模块加阴影，使之看上去有立体感（图 9-13）。

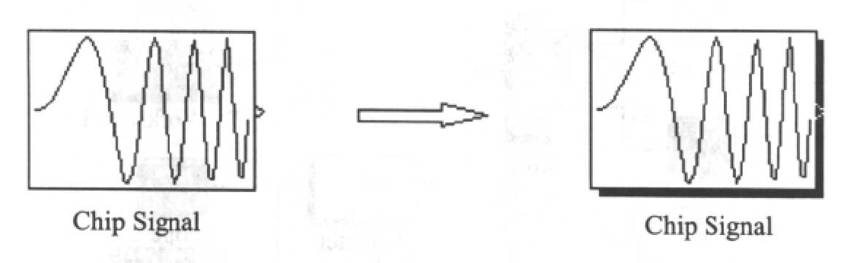


图 9-13　模块加阴影

2. 调整颜色

选择 Format 菜单中的 Foreground color 菜单项用于调整模块的前景颜色，包括模块上图案的颜色、模块名称的颜色及阴影的颜色。选择 Background color 菜单项用于选择模块背景颜色，指的是模块表面的颜色。调整屏幕颜色要选择 Screen color（屏幕颜色）菜单项，屏幕颜色指的是模型窗口的颜色（图 9-14）。

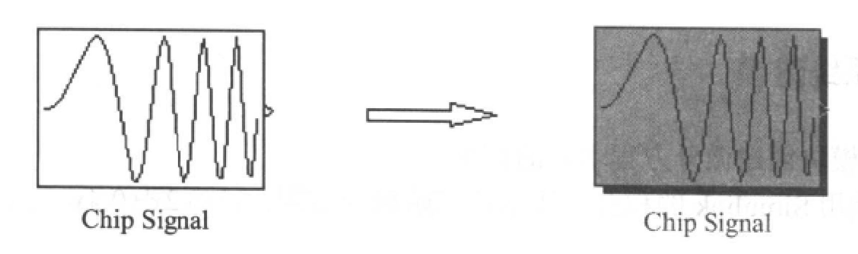
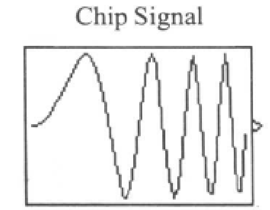
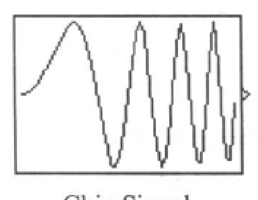


图 9-14　调整颜色后的效果

3. 改变模块名的显示位置

选择 Format 菜单中的 Flip name 可以将模块名换到对称的位置上，如将模块名从模块下方移到模块的上方，Hide name 选项可以将模块名隐藏起来不显示（图 9-15）。





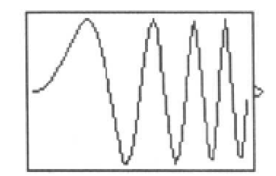


图 9-15　变换模块名的显示位置

4. 模块修饰的简单示例

创建一个简单系统，由信号发生器产生一个信号并叠加少量的随机噪声信号，数据分为两路，一路直接送至示波器显示，并将数据保存在工作空间，另一路经过传递函数计算后再送示波器显示，显示的同时也存储到工作空间。为了使创建的模型看起来更生动、更清晰，添加了一些修饰效果：(1) 加标题以对所创建的模型进行说明；(2) 模块加阴影，增加视觉效果；(3) 用颜色对模块的作用进行区分，以便于理解和检查。图 9-16 为修饰后的模型。

上述修饰将不影响模块的运行。

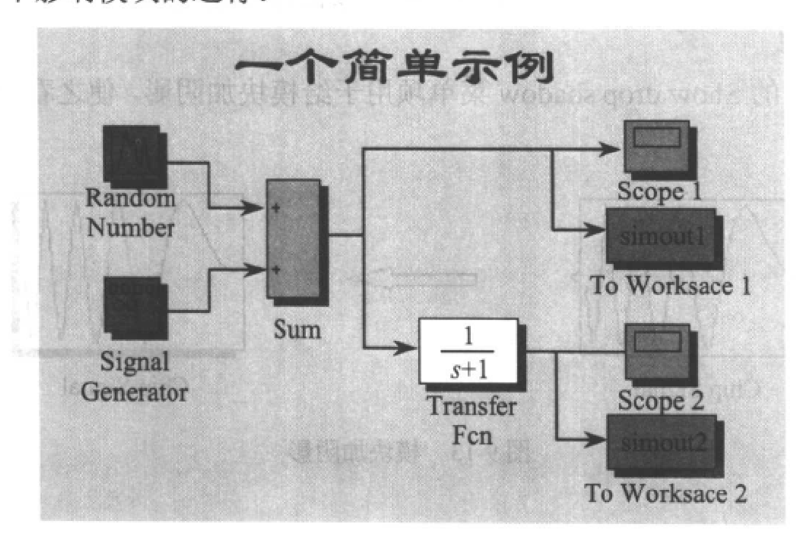


图 9-16　模型的修饰

9.4　仿真计算与分析方法

9.4.1　连续系统建模

1. 用积分模块创建微分方程求解的模型

【例 9-1】　使用 Simulink 的积分模块求解二阶微分方程：$x''+0.2x'+0.4x=0.2u(t)$ ，$u(t)$是单位阶跃函数。

本例将演示如何用积分器直接创建求解该微分方程的模型，步骤如下：

(1) 改写微分方程为：$x''=0.2u(t)-0.2x'-0.4x$；

(2) 利用 Simulink 库中的标准模块创建模型（图 9-17）。

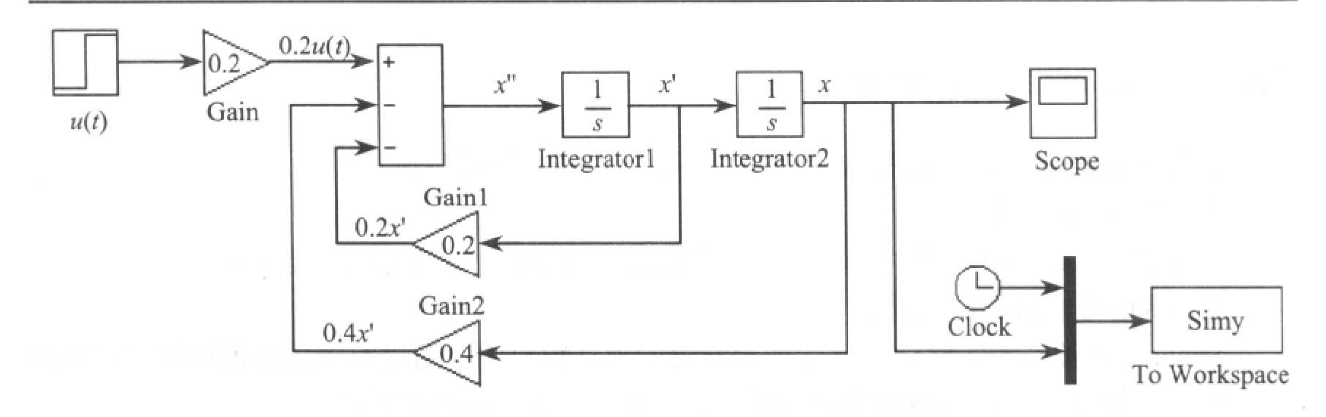


图 9-17 求解微分方程的积分模型

利用积分模块创建微分方程求解模型的核心思想是：x''经积分后得到 x'，再经积分得到 x，x'和 x 经代数运算产生 x''。

计算结果送到示波器显示，同时又通过 To workspace 模块以选定的矩阵方式存储在工作空间变量 *simy* 中。图 9-18 为示波器的显示结果。

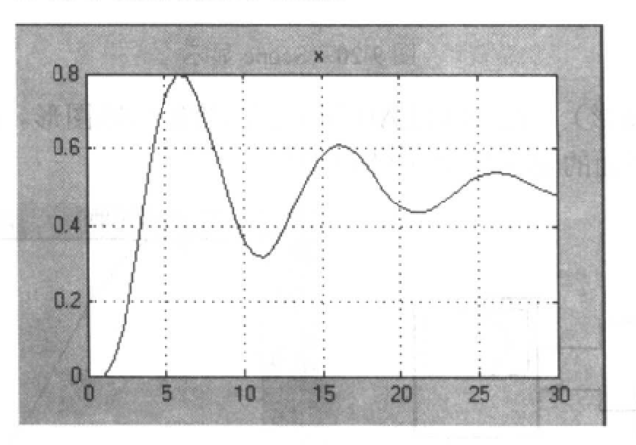


图 9-18 微分方程的求解结果

2. 用传递函数模块求解微分方程

【例 9-2】 使用 Simulink 的传递函数模块求解二阶微分方程：$x''+0.2x'+0.4x=0.2u(t)$，设初始状态为 0，$u(t)$是单位阶跃函数。

对方程两边进行 Laplace 变换，得到：

$$s^2X(s)+0.2sX(s)+0.4X(s)=0.2U(s)$$

整理后得到：

$$G(s)=X(s)/U(s)=0.2/(s^2+0.2s+0.4)$$

用传递函数模块建立求解微分方程的模型（图 9-19）。

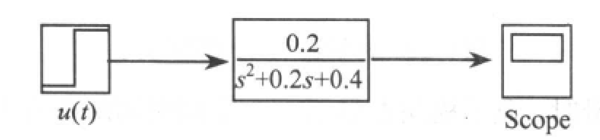


图 9-19 传递函数模型

9.4.2　Simulink 结果的分析

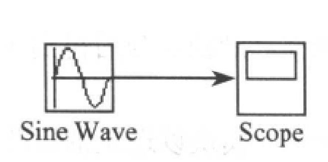
执行 Simulink 后，必须检查输出结果并做进一步的分析与判断。

1. 输出信号的观察

在 Sinks 函数库中，提供了多种观察输出信号的方法，大致可以分为 3 种类型。

(1) 将信号输出到显示模块

Scope（示波器）：将信号显示在示波器的独立窗口中。通过双击显示模块打开示波器显示窗口（图 9-20），示波器窗口可以放大、缩小，也可以将波形打印出来。



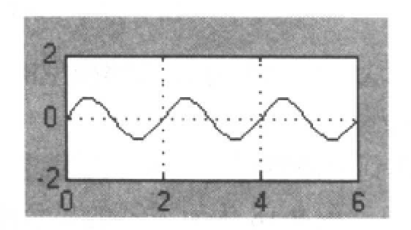
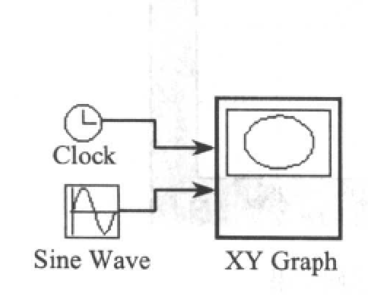


图 9-20　Scope 显示

XY Graph（XY 图形）：在 MATLAB 图形窗口绘制二维图形。该显示器有两个输入端，上面的输入作为 x，下面的输入作为 y（图 9-21）。



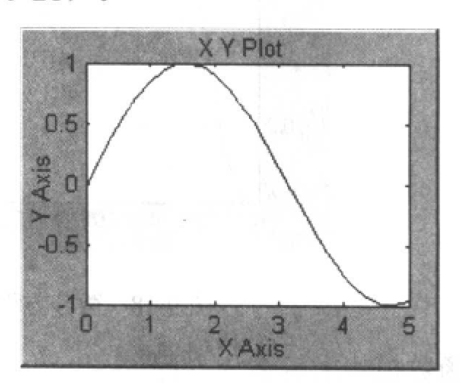


图 9-21　XY 图形显示

Display（数字显示）：将结果以数字形式显示出来。数字显示模块没有独立的显示窗口，只是在模块上的显示框中直接滚动显示数据结果（图 9-22）。

当输入数据是标量时，显示模块中只显示一个窗口。

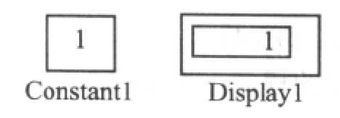


图 9-22　标量输入时的数字显示

当输入数据是行矢量时，显示模块在显示一个窗口时同时在右下角显示一个向右的小箭头，将模块向右拉开可以显示出多个窗口（图 9-23）。

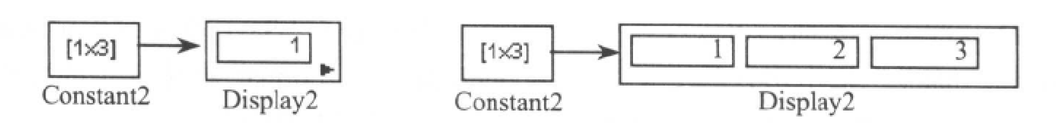


图 9-23 行矢量输入时的数字显示

当输入数据是列矢量时，显示模块在显示一个窗口时同时在右下角显示一个向下的小箭头，将模块向下拉开可以显示出多个窗口（图 9-24）。

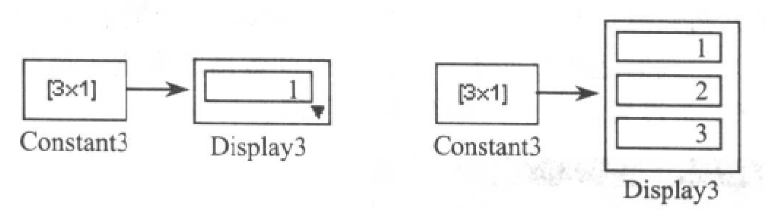


图 9-24 列矢量输入时的数字显示

当输入数据是矩阵时，显示模块在显示一个窗口时同时在右下角显示一个向右的和一个向下的小箭头，将模块向右下方拉开可以显示出多个窗口（图 9-25）。

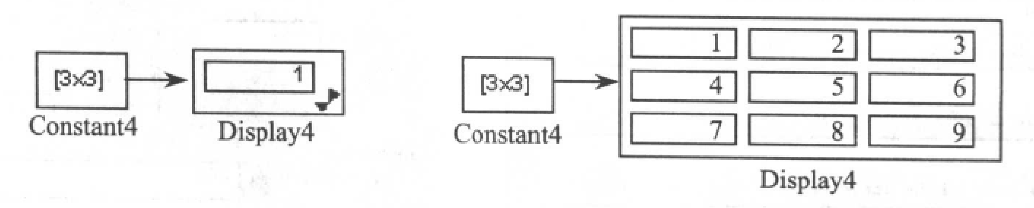


图 9-25 矩阵输入时的数字显示

只要将这 3 种显示器中的任意一种放在系统的输出端上，就可以直接观察仿真时的结果。

(2) 将仿真结果存储到工作空间，再用绘图命令在 MATLAB 命令窗口绘制图形。

存储方式有 3 种，可以根据具体情况选择使用。

第一种方法是通过示波器模块向工作空间存储数据。当使用示波器观察数据时，缺省情况下会将一个名为 ScopeData 的数据结构和名为 tout 的数组存储在工作空间中，通过下面的方法也可以选择用其他的数据类型存储数据或不通过示波器存储。在 Scope 窗口的工具栏中选择 Parameters（参数设置）按钮，打开一个对话框，见图 9-26。

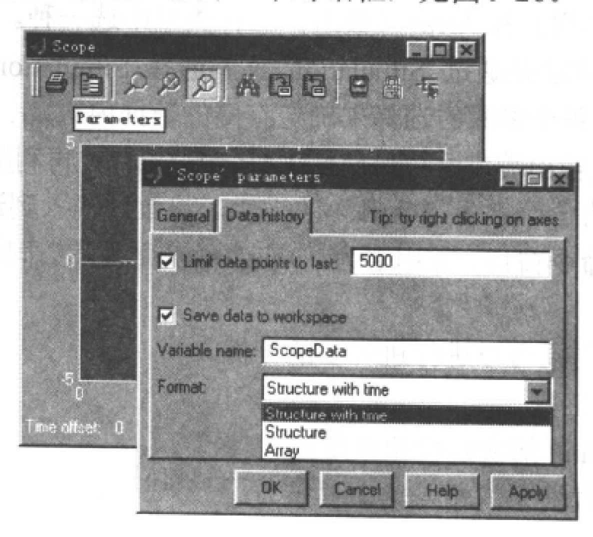


图 9-26 示波器参数设置对话框

选择 Data history 页面，可以看到在缺省状态下是勾选了“Save data to workspace”项，在“Variable name”框中指定的变量名是 ScopeData，在 Format 框中选定的数据格式是“Structure with time”（带有时间的数据结构）。变量名可以根据自己的需要设置，数据格式有“Structure with time”（带有时间的数据结构）、Structure（数据结构）和 Array（数组）三种可以选择，如果不希望从 Scope 中存储数据，可以去掉该项的勾选。

另一种方法是通过选择 Sinks 函数库中的 To workspace 模块，只要将数据输入到这个模块内，这个模块就会将数据保存到工作空间中的“simout”变量中（变量名可以自定义，缺省格式为数据结构），同时还可以产生一个存放时间数据的变量（缺省为 tout），双击 To workspace 模块打开参数设置对话框，如图 9-27 所示。



图 9-27　To workspace 模块的参数设置对话框

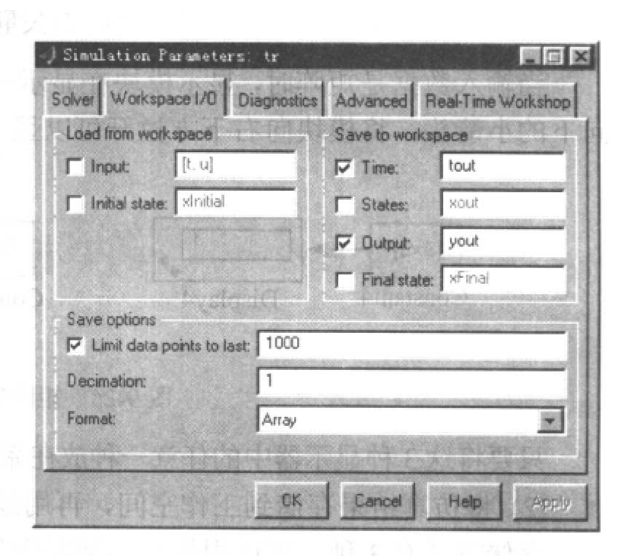


图 9-28　Simulation Parameters 菜单项中的 Workspace I/O 页

从对话框中可以看到，当缺省设置时数据以 Structure 格式存储在名为 simout 的变量中。变量名可以根据自己的需要改变，数据格式可以在“Structure With Time”（带时间的数据结构）、Structure（数据结构）和 Array（数组）三种类型中选一种。

第三种存储数据的方法是通过 Simulation 菜单选择 Simulation Parameters 菜单项中的 Workspace I/O 页，根据各个参数的选择来确定存储的数据内容及类型。见图 9-28。

(3) 将仿真结果通过输出端口返回到 MATLAB 命令窗口，再利用绘图命令绘出输出图形。

在 Sinks 函数库中有一个名为“Out1”的模块，将数据输入到这个模块，该模块就会将数据输出到 MATLAB 命令窗口，并用名为“yout”的变量保存，同时还将时间数据用“tout”变量保存。

存储在工作空间的结果可以进行进一步的分析。

2. 使用一般的分析工具

【例 9-3】　使用 Simulink 解决线性化系统问题。

状态空间所描述的线性系统输入输出关系为：

$$x' = Ax+Bu$$

$y = Cx+Du$

其中 x 代表状态矢量，u 代表输入矢量，y 代表输出矢量。

创建系统模型（图 9-29），并保存为“lmod.mdl”。

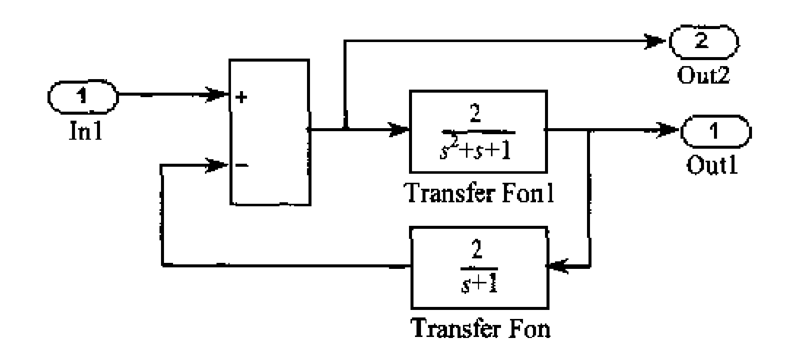


图 9-29 用于线性化的系统模型

在 MATLAB 命令窗口输入命令：

```
>>[A B C D]=linmod('lmod')
```

经过计算，可以得到系统线性化的状态空间矩阵：

```
A =
   -1       -1       -2
    1        0        0
    0        2       -1
B=
    1
    0
    0
C =
    0        2        0
    0        0       -2
D =
    0
    1
```

【例 9-4】 使用 Simulink 将线性化系统转成 LTI 对象。

使用 Simulink 可以将上面的线性化系统转成 LTI 对象，命令格式为：

```
>> sys=ss(A,B,C,D)
a =
       x1  x2  x3
  x1   -1  -1  -2
  x2    1   0   0
  x3    0   2  -1
b =
       u1
  x1    1
  x2    0
```

```
   x3   0
 c =
       x1  x2  x3
   y1   0   2   0
   y2   0   0  -2
 d =
        u1
   y1   0
   y2   1
Continuous-time model.
```

【例 9-5】　使用 Simulink 绘制波德图。

波德图是经常使用的一种图形，Simulink 中绘制波德图的命令格式为：

- bode(*A*, *B*, *C*, *D*)
- bode(*sys*)

图 9-30 为所绘制的波德图。

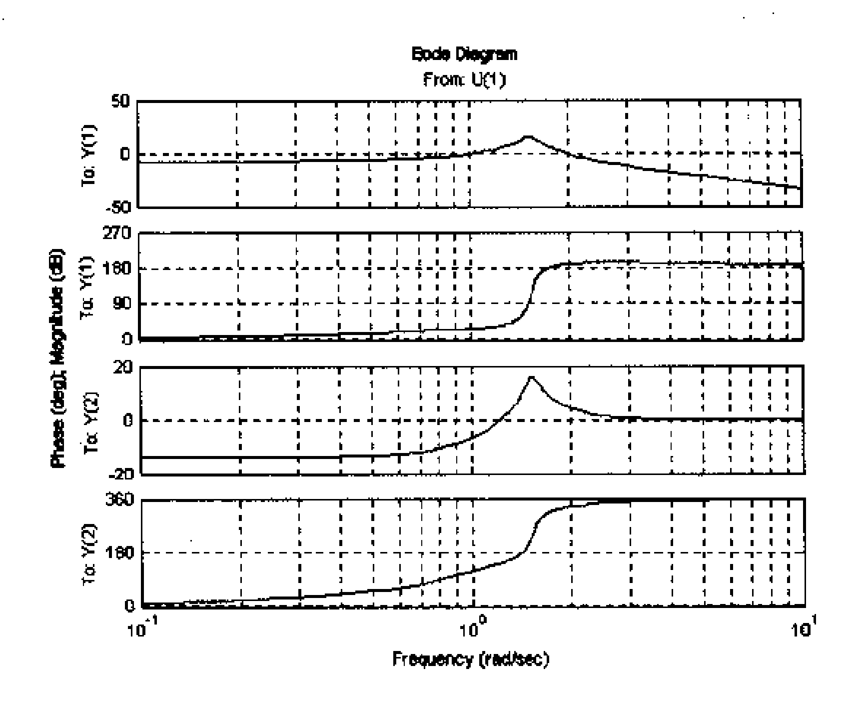


图 9-30　将空间矩阵绘成波德图

【例 9-6】　使用 Simulink 实现线性时间响应。

产生一个阶跃信号：

- step(*A*, *B*, *C*, *D*)
- step(*sys*)

图 9-31 为所绘制的阶跃信号图。

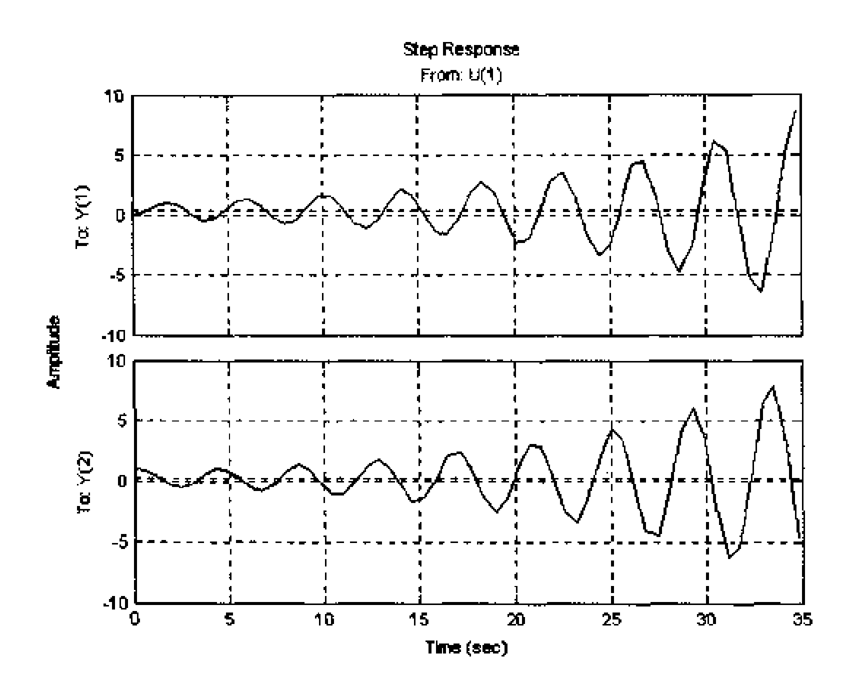


图 9-31 绘制阶跃信号图

产生一个脉冲信号：

- impulse(A, B, C, D)
- impulse(sys)

图 9-32 为所绘制的脉冲信号图。

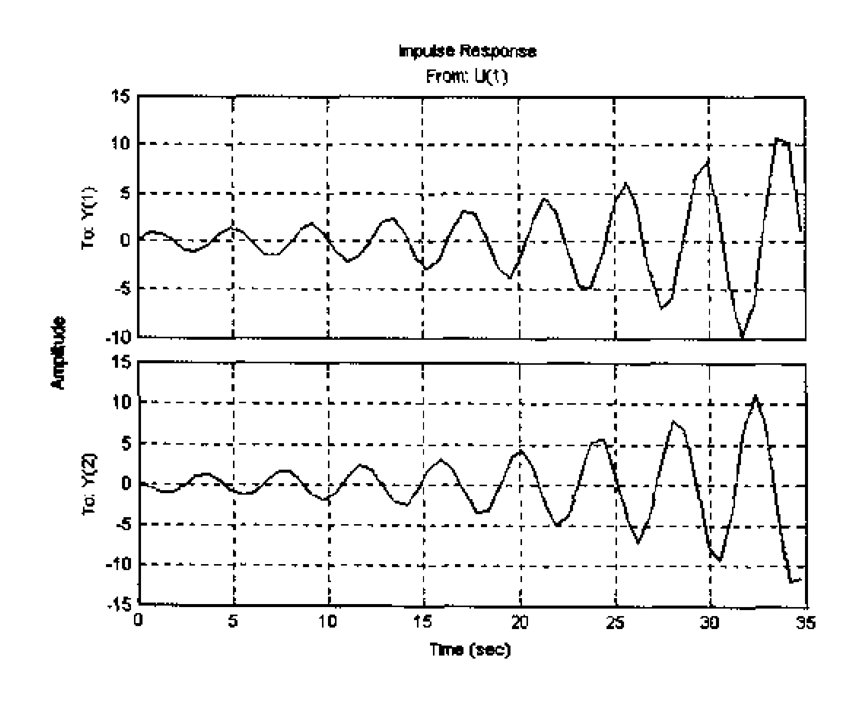


图 9-32 绘制脉冲信号图

【例 9-7】 使用 Simulink 求系统平衡点。

在非线性系统中，分析评估系统稳定性或稳定状态时大多需要计算平衡点，所谓平衡点

是指所有状态导数等于零的点，若仅部分状态导数等于零，则成为偏平衡点。

当要使输出为 1，并找出输入以及状态值时，可以利用 trim 函数实现。

仍以前面创建的“lmod”模型为例。

第一步：对状态变量 x 以及输入 u 做初步设定，并设定想要的输出值：

```
>> x=[0;0;0];
>> u=0;
>> y=[1;1];
```

第二步：使用索引变量确定哪些值可变，哪些是固定不变的：

```
>> ix=[ ];                % 表示状态不固定
>> iu=[ ];                % 表示输入不固定
>> iy=[1,2];              % 固定第一个输出及第二个输出
```

第三步：调用 trim 函数，求出系统平衡点：

```
>> [x,u,y,dx]=trim('lmod',x,u,y,ix,iu,iy)
x =
      0.0000
      0.66667
      1.3333
u =
      3.3333
y =
      1.3333
      0.66667
dx =
 -1.1102e-016
           0.0000
                0
```

习　题　9

9.1　什么是 Simulink？

9.2　如何进行下列操作：

(1) 翻转模块

(2) 给模型窗口加标题

(3) 指定仿真时间

(4) 设置示波器的显示刻度

9.3　有如下传递函数的控制系统，用 Simulink 建立系统模型，并对系统的阶跃响应进行仿真。

$$G(s)=\frac{1}{s^2+4s+8}$$

9.4　建立一个简单模型，用信号发生器产生一个幅度为 2 V、频率为 0.5 Hz 的正弦波，并叠加一个 0.1 V 的噪声信号，将叠加后的信号显示在示波器上并传送到工作空间。

9.5　建立一个简单模型，产生一组常数（1×5），再将该常数与其 5 倍的结果合成一个二维数组，用数字显示器显示出来。

9.6 建立一个模拟系统，将摄氏温度转换为华氏温度（T_f=9/5T_c+32）。

9.7 建立二阶系统 $G(S)=\dfrac{\omega_n}{S^2+2\zeta\omega_n S+\omega_n^2}$ 的脉冲响应模型，设 ω_n=10 Hz，观察当 0< ζ<1、 ζ=0、ζ=1 及 ζ>1 时系统的响应。

9.8 以 15 米/秒的速度从 10 米高的地方抛下皮球，建立并显示球弹跳轨迹的模型。

附　录

附录 A　MATLAB 6.1 命令和函数索引

A.1　开发环境

A.1.1　启动和退出

命令	说明	命令	说明
exit	退出 MATLAB (同 quit)	matlabrc	MATLAB 运行 M文件
finish	MATLAB关闭M文件	quit	退出MATLAB
matlab	启动 MATLAB (UNIX 系统)	startup	启动时运行用户定义的M文件

A.1.2　命令窗口

命令	说明	命令	说明
clc	清除命令窗口	home	将光标移至命令窗口左上角
diary	将操作保存在文件中	more	设置命令窗口的分页输出
dos	执行 DOS 命令	notebook	在Microsoft Word 中打开M文件
format	设置输出数据的显示格式	unix	执行UNIX 命令

A.1.3　获得帮助

命令	说明	命令	说明
doc	在MATLAB的帮助浏览器中显示文件	info	显示MathWorks公司或产品信息
docopt	查找UNIX平台下的帮助文件目录	lookfor	搜索指定的关键词
help	在命令窗口中显示函数的帮助信息	support	启动 MathWorks 技术支持网页
helpbrowser	打开帮助浏览器	web	将帮助浏览器设定为某文件或网站
helpwin	在帮助浏览器中显示在线帮助主题	whatsnew	显示当前版本MATLAB或工具箱信息

A.1.4　工作空间、文件和搜索路径

1. 工作空间

命令	说明	命令	说明
assignin	工作空间中的变量赋值	pack	整理工作空间内存
clear	清除工作空间，释放系统内存	which	查找函数和文件
evalin	在工作空间运行字符串表达式	who, whos	列出工作空间变量
exist	检查可能存在的变量或文件	workspace	显示工作空间窗口
openvar	在数组编辑器中编辑变量		

2. 文件

命令	说明	命令	说明
cd	改变工作目录	matlabroot	返回MATLAB安装的根目录
copyfile	复制文件	mkdir	创建新目录
delete	删除文件或图形对象	pwd	显示当前目录
dir	显示目录列表	rehash	更新函数和文件系统缓存
exist	检查可能存在的变量或文件	type	列出文件
filebrowser	显示当前目录浏览器	what	列出当前目录下MATLAB 的指定文件
lookfor	搜索指定的关键词	which	查找函数和文件
ls	显示UNIX的目录列表		

3. 搜索路径

命令	说明	命令	说明
addpath	在MATLAB的搜索路径中增加路径	path	查看或改变MATLAB的搜索路径
genpath	产生路径字符串	pathtool	打开搜索路径对话框
partialpath	部分路径名	rmpath	从MATLAB的搜索路径中删除路径

A.1.5　编程工具

1. 编辑和调试

命令	说明	命令	说明
dbclear	清除断点	dbstep	从当前断点处执行一行或多行
dbcont	恢复运行	dbstop	在M函数中设置断点
dbdown	退出函数调试	dbtype	列出带行标号的M文件
dbquit	退出调试模式	dbup	进入函数调试
dbstack	显示函数调用堆栈	edit	编辑或创建M文件
dbstatus	列出全部断点	keyboard	在M文件中调用键盘

2. 代码控制

命令	说明	命令	说明
checkin	登录代码控制系统	customverctrl	使用自定义代码控制系统
checkout	退出代码控制系统	undocheckout	撤消前一次从代码控制系统退出
cmopts	返回代码控制系统名称		

3. 调用记录

命令	说明	命令	说明
profile	实现M文件的调用记录	profreport	生成调用记录报告

A.1.6　系统

命令	说明	命令	说明
computer	计算机信息	usejava	判断MATLAB是否已支持Java 特性
javachk	产生基于Java特性支持的错误信息	ver	显示版本信息
license	显示MATLAB的许可证号码	version	获得MATLAB 的版本号

A.1.7　性能优化工具和技术

命令	说明	命令	说明
memory	显示内存限制	rehash	更新函数和文件系统缓存
pack	整理工作空间内存	sparse	创建稀疏矩阵
profile	优化M文件代码的性能	zeros	创建零矢量
profreport	生成调用记录报告		

A.2　数学

A. 2. 1　数组和矩阵

1. 基本信息

命令	说明	命令	说明
disp	显示数组	issparse	当矩阵为稀疏矩阵时值为真
display	显示数组	length	矢量的长度
isempty	当矩阵为空矩阵时值为真	ndims	维数
isequal	当数组数值相等时值为真	numel	元素数
islogical	当数组为逻辑数组时值为真	size	矩阵尺寸
isnumeric	当数组为数值数组时值为真		

2. 运算

命令	说明	命令	说明
+	加	/	矩阵右除或斜线
+	一元加	'	转置，共轭转置
-	减	.'	非共轭转置
-	一元减	.*	数组乘法
*	矩阵乘法	.^	数组幂
^	矩阵幂	.\	数组左除
\	矩阵左除或反斜线	./	数组右除

3. 基本操作

命令	说明	命令	说明
:(冒号)	数组的索引号，数组重新排列	kron	Kronecker张量乘
blkdiag	分块对角阵	max	数组的最大元素
cat	连接	min	数组的最小元素
cross	矢量叉乘	permute	多维数组的重排序
cumprod	元素累计积	prod	数组元素积
cumsum	元素累计和	repmat	填充数组
diag	建立或提取对角阵	reshape	数组重新排列
dot	矢量点乘	rot90	矩阵旋转90°
end	最大下标	sort	按升序排列元素
find	查找数组非零元素的下标和值	sortrows	按升序排列行
fliplr	矩阵左右翻转	sum	数组元素求和
flipud	矩阵上下翻转	sqrtm	矩阵平方根
flipdim	沿指定维数翻转矩阵	sub2ind	下标的线性表示转为多维表示
horzcat	水平连接	tril	矩阵的下三角部分
ind2sub	下标的多维表示转为线性表示	triu	矩阵的上三角部分
ipermute	多维数组的逆排列维	vertcat	垂直连接

4. 基本矩阵和数组

命令	说明	命令	说明
:(冒号)	规则间隔的矢量	meshgrid	产生三维绘图网格
blkdiag	分块对角阵	ndgrid	产生用于多维函数和插值的数组
diag	建立或提取对角阵	ones	创建全"1"矩阵
eye	单位矩阵	rand	均匀分布的随机数和数组
freqspace	频率响应的频率间隔	randn	正态分布的随机数和数组
linspace	产生线形间隔的矢量	repmat	填充数组
logspace	产生对数间隔的矢量	zeros	创建全零矩阵

5. 特殊矩阵

命令	说明	命令	说明
compan	伴随矩阵	magic	魔方矩阵
gallery	测试矩阵	pascal	Pascal 矩阵
hadamard	Hadamard 矩阵	rosser	经典对称特征值测试问题
hankel	Hankel 矩阵	toeplitz	Toeplitz 矩阵
hilb	Hilbert 矩阵	vander	Vandermonde 矩阵
invhilb	逆 Hilbert 矩阵	wilkinson	Wilkinson's 特征值测试矩阵

A.2.2　线性代数

1. 矩阵分析

命令	说明	命令	说明
cond	计算矩阵的条件数	orth	正交化
condeig	按特征值计算矩阵的条件数	rank	矩阵的秩
det	计算行列式值	rcond	矩阵的逆条件数估计
norm	计算矩阵或矢量范数	rref	缩减行格式矩阵
normest	估计矩阵的2-范数	subspace	子空间之间的夹角
null	零矩阵	trace	对角元素的和

2. 线性方程

命令	说明	命令	说明
\ 和 /	线性方程求解	lscov	存在已知协方差的最小二乘解
chol	Cholesky 分解	lsqnonneg	非负最小二乘
cholinc	不完全的 Cholesky 分解	lu	LU 矩阵分解
cond	计算矩阵的条件数	luinc	不完全的 LU 分解
condest	范数条件数估计	pinv	Moore-Penrose 矩阵伪逆
funm	运行通用矩阵函数	qr	正交三角阵分解
inv	矩阵求逆	rcond	矩阵逆条件数估计

3.特征值和奇异值

命令	说明	命令	说明
balance	矩阵均衡以提高特征值计算精度	poly	指定根的多项式
cdf2rdf	变复对角矩阵为实分块对角形式	polyeig	多项式特征值问题
condeig	按特征值计算矩阵的条件数	qz	广义特征值的QZ 分解
eig	特征值和特征矢量	rsf2csf	变实分块对角形式为复分块对角形式
eigs	稀疏矩阵的特征值和特征矢量	schur	Schur 分解
gsvd	广义奇异值分解	svd	奇异值分解
hess	矩阵的Hessenberg形式	svds	稀疏矩阵的奇异值和矢量

4.矩阵的对数和指数

命令	说明	命令	说明
expm	矩阵指数	sqrtm	矩阵平方根
logm	矩阵对数		

5.因式分解

命令	说明	命令	说明
balance	矩阵均衡以提高特征值计算精度	planerot	给定平面旋转
cdf2rdf	变复对角矩阵为实分块对角形式	qr	正交三角阵分解
chol	Cholesky 分解	qrdelete	在正交三角阵分解中删除列
cholinc	不完全的 Cholesky 分解	qrinsert	在正交三角阵分解中插入列
cholupdate	秩为1的Cholesky 分解	qrupdate	秩为1的正交三角阵分解
lu	LU 矩阵分解	qz	广义特征值的正交三角阵分解
luinc	不完全的LU 矩阵分解	rsf2csf	变实分块对角形式为复分块对角形式

A.2.3 基本数学函数

1.三角函数

命令	说明	命令	说明
acos, acosh	反余弦和反双曲余弦	cos, cosh	余弦和双曲余弦
acot, acoth	反余切和反双曲余切	cot, coth	余切和双曲余切
acsc, acsch	反余割和反双曲余割	csc, csch	余割和双曲余割
asec, asech	反正割和反双曲正割	sec, sech	正割和双曲正割
asin, asinh	反正弦和反双曲正弦	sin, sinh	正弦和双曲正弦
atan, atanh	反正切和反双曲正切	tan, tanh	正切和双曲正切
atan2	四象限反正切		

2.指数函数

命令	说明	命令	说明
exp	指数	nextpow2	大于或等于变量的2 的下一次幂
log	自然对数	pow2	2 的指数
log2	以2为底的对数	sqrt	平方根
log10	常用对数		

3.复数

命令	说明	命令	说明
abs	幅值	imag	复矩阵虚部
angle	相角	isreal	当数组为实数时为真
complex	用实部和虚部构造复数	j	虚数单位
conj	复共轭函数	real	复矩阵实部
cplxpair	将数分为共轭对	unwrap	展开相角函数
i	虚数单位		

4.舍入和剩余函数

命令	说明	命令	说明
fix	朝零方向舍入	mod	（带符号）求余函数
floor	朝负方向舍入	rem	无符号求余函数
ceil	朝正方向舍入	sign	符号函数
round	四舍五入函数		

5. 离散数学

命令	说明	命令	说明
factor	数的因式分解	nchoosek	组合运算
factorial	阶乘	perms	排列运算
gcd	最大公因数	primes	产生质数
isprime	如为质数为真	rat, rats	有理逼近
lcm	最小公倍数		

A.2.4　数据分析和傅里叶变换

1. 基本操作

命令	说明	命令	说明
cumprod	元素累计积	prod	元素积
cumsum	元素累计和	sort	按升序排列元素
cumtrapz	梯形法累积数值积分	sortrows	按升序排列行
max	最大元素	std	标准差
mean	平均值	sum	元素和
median	中值	trapz	梯形法求数值积分
min	最小元素	var	方差（标准差的平方）

2. 有限差分

命令	说明	命令	说明
del2	离散 Laplacian变换	gradient	近似梯度
diff	差分和近似微分		

3. 相关

命令	说明	命令	说明
corrcoef	相关系数	subspace	子空间之间的夹角
cov	协方差矩阵		

4. 滤波和卷积

命令	说明	命令	说明
conv	卷积和多项式乘法	detrend	移除数据中的线性趋势
conv2	二维卷积	filter	一维数字滤波器
convn	N维卷积	filter2	二维数字滤波器
deconv	反卷积和多项式除法		

5. 傅里叶变换

命令	说明	命令	说明
abs	绝对值和复数模	ifft	一维快速傅里叶逆变换
angle	相角	ifft2	二维快速傅里叶逆变换
fft	一维快速傅里叶变换	ifftn	N维离散傅里叶逆变换
fft2	二维快速傅里叶变换	ifftshift	快速傅里叶逆变换平移
fftn	N维离散傅里叶变换	nextpow2	大于或等于变量的2 的下一次幂
fftshift	将零点平移到频谱中心	unwrap	修正相角

A.2.5　多项式

命令	说明	命令	说明
conv	卷积和多项式乘法	polyint	多项式积分
deconv	反卷积和多项式除法	polyval	多项式计算
poly	指定根的多项式	polyvalm	带矩阵的多项式计算
polyder	多项式微分	residue	留数计算
polyeig	多项式特征值问题	roots	多项式根
polyfit	多项式曲线拟合		

A.2.6　插值和计算几何

1. 插值

命令	说明	命令	说明
dsearch	最近点搜索	interpn	多维插值（查表）
dsearchn	多维最近点搜索	meshgrid	产生三维绘图网格
griddata	数据栅格和曲线拟合	mkpp	分解多项式
griddata3	三维数据的数据栅格和超曲线拟合	ndgrid	产生用于多维函数和插值的数组
griddatan	多维数据的数据栅格和超曲线拟合	pchip	分段三次Hermite插值多项式
interp1	一维插值（查表）	ppval	分段多项式计算
interp2	二维插值（查表）	spline	三次样条插值
interp3	三维插值（查表）	tsearchn	多维最近点单一搜索
interpft	使用快速傅里叶变换方法一维插值	unmkpp	分解多项式过程

2. Delaunay三角分解

命令	说明	命令	说明
delaunay	Delaunay 三角分解	trimesh	三角形网格绘图
delaunay3	三维Delaunay三角分解	triplot	二维三角形网格绘图
delaunayn	多维Delaunay三角分解	trisurf	三角形曲面绘图
dsearch	最近点搜索	tsearch	最接近的Delaunay三角搜索
dsearchn	多维最近点搜索	tsearchn	最接近的多维Delaunay三角搜索
tetramesh	四面体网格绘图		

3. 凸面图

命令	说明	命令	说明
convhull	凸面图	plot	二维绘图
convhulln	多维凸面图	trisurf	三角形曲面绘图
patch	创建块状绘图对象		

4. Voronoi 图表

命令	说明	命令	说明
dsearch	最近点搜索	voronoi	生成Voronoi 图表
patch	创建块状绘图对象	voronoin	多维Voronoi 图表
plot	二维绘图		

5. 域生成

命令	说明	命令	说明
meshgrid	产生三维绘图网格	ndgrid	产生用于多维函数和插值的数组

A.2.7　坐标变换

命令	说明	命令	说明
cart2sph	转换笛卡儿坐标为球坐标	pol2cart	转换极坐标为笛卡儿坐标
cart2pol	转换笛卡儿坐标为极坐标	sph2cart	转换球坐标为笛卡儿坐标

A.2.8　非线性数学方法

1. 常微分方程

命令	说明	命令	说明
deval	解微分方程问题	ode23t	解适度刚性微分方程，梯形法则
ode113	解非刚性微分方程，变阶方法	ode23tb	解刚性微分方程，低阶方法
ode15s	解刚性微分方程，变阶方法	ode45	解非刚性微分方程，变阶方法
ode23	解非刚性微分方程，低变阶方法	odeget	得到常微分方程参数
ode23s	解刚性微分方程，低阶方法	odeset	创建改变参数结构

2. 边界值问题

命令	说明	命令	说明
bvp4c	解常微分方程的两点边界值问题	bvpget	得到边界值参数
bvpset	创建改变边界值参数结构	deval	解微分方程问题

3. 偏微分方程

命令	说明	命令	说明
pdepe	解椭圆偏微分方程的初始-边界值问题	pdeval	pdepe计算的解的估计

4. 优化

命令	说明	命令	说明
fminbnd	函数的下界	lsqnonneg	非负约束的最小二乘
fminsearch	多维无约束非线性最小化	optimset	创建改变优化参数结构
fzero	寻找零点	optimget	从参数结构得到优化参数

5. 数值积分

命令	说明	命令	说明
quad	数值积分（低阶方法）	dblquad	数值两步积分
quadl	数值积分（高阶方法）		

A.2.9　专用数学函数

命令	说明	命令	说明
airy	Airy 函数	erf	误差函数
besselh	第三类Bessel 函数(Hankel 函数)	erfc	补充误差函数
besseli	改进的第一类Bessel 函数	erfcinv	补充误差逆函数
besselj	第一类Bessel 函数	erfcx	比例补充误差函数
besselk	改进的第二类Bessel 函数	erfinv	误差逆函数
bessely	第二类Bessel 函数	expint	指数积分
beta	Beta 函数	gamma	γ 函数
betainc	不完全的 beta 函数	gammainc	不完全 γ 函数
betaln	beta 函数的对数	gammaln	γ 函数的对数
ellipj	Jacobi 椭圆函数	legendre	联合 Legendre 函数
ellipke	第一类和第二类完全椭圆积分		

A.2.10　稀疏矩阵

1. 基本稀疏矩阵

命令	说明	命令	说明
spdiags	以对角阵形成稀疏矩阵	sprandn	稀疏正态分布随机矩阵
speye	稀疏单位矩阵	sprandsym	对称的稀疏随机矩阵
sprand	稀疏均匀分布随机矩阵		

2. 稀疏矩阵和满阵间的转换

命令	说明	命令	说明
find	查找数组非零元素的下标和值	sparse	建立稀疏矩阵
full	将稀疏矩阵转为满阵	spconvert	从稀疏矩阵的外部形式输入

3. 稀疏矩阵操作

命令	说明	命令	说明
issparse	如是稀疏矩阵为真	spfun	非零矩阵元素的应用函数
nnz	非零元素的数目	spones	用1代替非零矩阵元素
nonzeros	非零矩阵元素	spparms	设置稀疏矩阵过程参数
nzmax	非零矩阵元素的内存分配	spy	稀疏图可视化
spalloc	稀疏矩阵的空间分配		

4. 重排算法

命令	说明	命令	说明
colamd	列近似最小度排列	randperm	随机排列
colmmd	列最小度排列	symamd	对称近似最小度排列
colperm	列排列	symmmd	对称最小度排列
dmperm	Dulmage-Mendelsohn 排列	symrcm	对称反 Cuthill-McKee 排列

5. 线性代数

命令	说明	命令	说明
cholinc	不完全 Cholesky 分解	normest	矩阵2 范数估计
condest	1范数条件数估计	sprank	结构秩
eigs	特征值和特征矢量	svds	稀疏矩阵的奇异值和奇异矢量
luinc	不完全 LU分解		

6. 线性方程（迭代法）

命令	说明	命令	说明
bicg	双共轭梯度法	minres	最小残差法
bicgstab	双共轭梯度稳定法	pcg	预处理共轭梯度法
cgs	共轭梯度平方法	qmr	准最小残差法
gmres	广义最小残差法	spaugment	形式最小二乘扩充系统
lsqr	正态方程的共轭梯度LSQR法	symmlq	对称 LQ 法

7. 树型操作

命令	说明	命令	说明
etree	消元树	symbfact	符号分解分析
etreeplot	画消元树图	treelayout	展示树或森林
gplot	绘图（以图论）	treeplot	绘制树图

A.2.11　数学常数

命令	说明	命令	说明
eps	精度相关的浮点数	NaN	非数
i	虚数单位	pi	圆周率, π
Inf	无穷大, ∞	realmax	最大正浮点数
j	虚数单位	realmin	最小负浮点数

A.3　程序设计和数据类型

A.3.1　数据类型

1. 数值变量

命令	说明	命令	说明
[ ]	构造数组	isnumeric	判断项是否为数值量
cat	连接数组	isreal	判断全部数组元素是否为实数
class	返回对象的类（即，数值）	permute	多维数组维数重排
find	查找数组非零元素的下标和值	reshape	数组重新排列
ipermute	反向置换数组维数	squeeze	消去次元
isa	判断对象是否为给定类（即，数值）	zeros	创建全零矩阵
isequal	判断数组是否数值相等		

2. 字符和字符串

MATLAB中字符串的描述

命令	说明	命令	说明
strings	描述MATLAB字符串句柄	strings	描述MATLAB字符串句柄

字符串的定义和操作

命令	说明	命令	说明
blanks	定义空字符串	sscanf	以确定格式读字符串
char	定义字符数组或字符串	strcat	字符串连接
cellstr	由字符数组定义字符串的单元数组	strjust	对齐字符串
datestr	转换为日期格式	strread	从字符串读数据
deblank	删除结尾空格	strrep	字符串搜索和置换
lower	转换为小写	strvcat	字符串纵向连接
sprintf	以确定格式写字符串	upper	转换为大写

比较和搜索字符串

命令	说明	命令	说明
class	返回对象的类（即字符串）	strcmp	比较字符串
findstr	搜索子串	strcmpi	比较字符串，忽略大小写
isa	判断对象是否为给定类（即字符串）	strfind	在另一字符串中查找字符串
iscellstr	判断是否为字符串单元	strmatch	查找字符串可能的匹配
ischar	判断是否为字符串	strncmp	比较字符串的前n 个字符
isletter	判断数组元素是否为字母	strncmpi	比较字符串前n 个字符，忽略大小写
isspace	判断元素是否为空格	strtok	返回字符串的第一个连续串

字符串表达式求值

命令	说明	命令	说明
eval	运行字符串表达式	evalin	在工作空间运行字符串表达式
evalc	运行字符串表达式		

3. 结构

命令	说明	命令	说明
cell2struct	转换单元数组为结构数组	isfield	判断项是否为结构域
class	返回对象的类（即结构）	isstruct	判断项是否为结构数组
deal	将输入处理为输出	rmfield	消除结构域
fieldnames	结构域名	setfield	设置结构域目录
getfield	得到结构的域目录	struct	创建结构数组
isa	判断对象是否为给定类（即结构）	struct2cell	转换结构数组为单元数组
isequal	判断数组是否数值相等		

4. 单元数组

命令	说明	命令	说明
{}	构造单元数组	deal	将输入处理为输出
cell	构造单元数组	isa	判断对象是否为给定类（即单元）
cellfun	用单元函数操作单元数组	iscell	判断项是否为单元数组
cellstr	有字符数组创建单元数组	iscellstr	判断项是否为字符串的单元数组
cell2struct	转换单元数组为结构数组	isequal	判断数组是否数值相等
celldisp	显示单元数组内容	num2cell	转换数值数组为单元数组
cellplot	单元数组结构的图形显示	struct2cell	转换结构数组为单元数组
class	返回对象的类（即单元）		

5. 数据类型转换

数值型

命令	说明	命令	说明
double	转换为双精度	single	转换为单精度
int8	转换为8位符号整数	uint8	转换为8位无符号整数
int16	转换为16位符号整数	uint16	转换为16位无符号整数
int32	转换为32位符号整数	uint32	转换为32位无符号整数

字符串转为数值型

命令	说明	命令	说明
base2dec	转换N进制字符串为十进制数	hex2num	转换十六进制字符串为数
bin2dec	转换二进制字符串为十进制数	str2double	转换字符串为双精度数
hex2dec	转换十六进制字符串为十进制数	str2num	转换字符串为数

数值型转为字符串

命令	说明	命令	说明
char	转换为字符（字符串）	int2str	转换整数为字符串
dec2base	转换十进制数为N进制字符串	mat2str	转换矩阵为字符串
dec2bin	转换十进制数为二进制进制字符串	num2str	转换数为字符串
dec2hex	转换十进制数为十六进制字符串		

其他转换

命令	说明	命令	说明
cell2struct	转换单元数组为结构数组	num2cell	转换数值数组为单元数组
datestr	转换日期为字符串	str2func	转换函数名字符串为函数句柄
func2str	转换函数句柄为函数名字符串	struct2cell	转换结构数组为单元数组
logical	转换数值数组为逻辑数组		

确定数据类型

命令	说明	命令	说明
is*	判断状态	isjava	判断项是否为Java 对象
isa	判断对象是否为给定的类或Java类	islogical	判断项是否为逻辑数组
iscell	判断是否为单元数组	isnumeric	判断项是否为数值数组
iscellstr	判断是否为字符串单元数组	isobject	判断项是否为MATLAB OOPs 对象
ischar	判断是否为字符数组	isstruct	判断项是否为MATLAB 结构数组
isfield	判断域是否在结构数组中		

A.3.2 数组

1. 数组操作

命令	说明	命令	说明
[ ]	构造数组	-	减法和一元减法
,	分隔数组行元素	.*	数组乘法
;	分隔数组列元素	./	数组右除
:	数组元素的给定范围	.\	数组左除
end	数组的结尾	.^	数组乘方
+	加法和一元加法	.'	数组转置

2. 数组基本信息

命令	说明	命令	说明
disp	显示文本或数组	islogical	判断项是否为逻辑数组
display	显示文本或数组	length	矢量长度
isempty	判断数组是否为空数组	ndims	数组维数
isequal	判断数组是否数值相等	numel	矩阵或单元数组的单元数
isnumeric	判断项是否为数值数组	size	数组尺寸

3. 数组操作

命令	说明	命令	说明
:	数组元素的给定范围	permute	多维数组维数重排
blkdiag	由输入参数构造块状对角阵	repmat	填充数组
cat	连接数组	reshape	数组重新排列
find	查找数组非零元素的下标和值	rot90	矩阵旋转90度
fliplr	矩阵左右翻转	shiftdim	替换维数
flipud	矩阵上下翻转	sort	按升序排列元素
flipdim	沿指定维数翻转矩阵	sortrows	按升序排列行
horzcat	水平连接	squeeze	消去次元
ind2sub	下标的线性表示转为多维表示	sub2ind	下标的多维表示转为线性表示
ipermute	多维数组维数反向重排	vertcat	垂直连接

4. 基本数组

命令	说明	命令	说明
:	规则间隔矢量	ndgrid	产生用于多维函数和插值的数组
blkdiag	由输入参数构造块状对角阵	ones	创建全"1"矩阵
eye	单位矩阵	rand	均匀分布的随机数和数组
linspace	产生线性空间矢量	randn	正态分布的随机数和数组
logspace	产生对数空间矢量	zeros	创建全零矩阵
meshgrid	产生三维绘图网格		

A.3.3　运算符和运算

1. 特殊符号

命令	说明	命令	说明
:	数组元素的给定范围	,	数组行元素分隔符
()	函数的参数选项、操作次序	;	数组列元素分隔符
[]	构造数组	%	注释行
{}	构造单元数组	!	执行操作系统命令
.	小数点、结构域分隔符	=	赋值
...	续行号		

2. 算术操作

命令	说明	命令	说明
+	加	^	矩阵幂
-	减	'	矩阵转秩
.	小数点	.*	数组乘法
=	赋值	./	数组右除
*	矩阵乘法	.\	数组左除
/	矩阵右除	.^	数组幂
\	矩阵左除	.'	数组转置

3. 位方式操作

命令	说明	命令	说明
bitand	位方式与	bitset	在指定点设置位
bitcmp	位方式补	bitshift	位方式移位
bitor	位方式或	bitget	在指定点取位
bitmax	最大浮点整数	bitxor	位方式异或

4. 关系操作

命令	说明	命令	说明
<	小于	>=	大于或等于
<=	小于或等于	==	等于
>	大于	~=	不等于

5. 逻辑操作

命令	说明	命令	说明
&	逻辑与	is*	判断状态
|	逻辑或	isa	判断对象是否为给定类
~	逻辑非	iskeyword	判断字符串是否为MATLAB关键词
all	测试全部元素是否非零	isvarname	判断字符串是否为有效的变量名
any	测试是否存在非零元素	logical	将数值量转换为逻辑量
find	查找数组非零元素的下标和值	xor	逻辑异或

6. 集合操作

命令	说明	命令	说明
Intersect	两矢量的交集	setxor	两矢量的异或集
Ismember	判断集合成员	union	两矢量的并集
Setdiff	两矢量的补集	unique	矢量的独立元素

7. 日期和时间

命令	说明	命令	说明
calendar	指定月的日历	datevec	日期组成
clock	以日期矢量方式返回当前时间	eomday	月底日
cputime	以秒返回 CPU 占用时间	etime	占用时间
date	以字符串方式返回当前日期	now	当前日期和时间
datenum	日期数字	tic, toc	秒表记时器
datestr	日期数字字符串	weekday	返回星期

A.3.4 MATLAB程序设计

1. 函数和命令的M文件

命令	说明	命令	说明
()	带入函数参数	nargin	函数的输入参数数量
%	注释行	nargout	函数的输出参数数量
...	续行号	nargchk	检查输入参数数量
depfun	查找M或P文件中的独立函数	nargoutchk	验证输出参数数量
depdir	查找M或P文件中的独立目录	pcode	生成P代码文件
function	定义函数	script	命令M文件
input	要求用户输入	varargin	带入参数
inputname	输入参数名	varargout	返回参数
mfilename	当前运行的M文件名		

2. 运行不等式和函数

命令	说明	命令	说明
builtin	使用重载方法运行内部函数	isvarname	判断字符串是否为有效的变量名
cellfun	在单元数组的每一元素中执行函数	pause	暂停运行
eval	运行字符串表达式	run	运行不在当前目录下的命令
evalc	运行字符串表达式	script	命令M文件
evalin	在工作空间运行表达式	symvar	判断表达式中的符号变量
feval	运行函数	tic, toc	秒表计时器
iskeyword	判断字符串是否为MATLAB关键词		

3. 内存中的变量和函数

命令	说明	命令	说明
assignin	工作空间中的变量赋值	mlock	防止从内存中清除M文件
global	定义全局变量	munlock	允许从内存中清除M文件
inmem	返回内存中的函数名	pack	整理工作空间内存
isglobal	判断变量是否为全局变量	persistent	定义永久变量
mislocked	M文件不能被清除时为真	rehash	刷新函数和文件系统缓存

4. 控制流

命令	说明	命令	说明
break	终止循环运行	for	重复执行指定次数
case	多选开关语句	if	条件运行语句
catch	检测语句	otherwise	开关语句的缺省部分
continue	运行下一次循环	return	返回调用函数
else	条件运行语句（配合if使用）	switch	多选开关语句
elseif	条件运行语句（配合if使用）	try	检测语句
end	终止条件语句，或指示最大下标	while	重复执行无限次数
error	显示错误信息		

5. 函数句柄

命令	说明	命令	说明
class	返回对象的类（即函数句柄）	func2str	从函数句柄构造函数名字符串
feval	运行函数	isa	判断对象是否为函数句柄
function_handle	描述函数句柄数据类型	isequal	判断函数句柄是否数值相等
functions	返回函数句柄信息	str2func	从函数名字符串构造函数句柄

6. 面对对象编程

MATLAB 类与对象

命令	说明	命令	说明
class	创建对象或返回对象的类	methodsview	显示类实现的方法信息
fieldnames	列出属于对象的公共类	saveobj	用户对象保存函数的扩展
inferiorto	建立子类关系	subsasgn	自定义对象使用下标的引用或赋值
isa	判断对象是否为给定类	subsindex	索引表达式中使用对象
isobject	判断是否为MATLAB面对对象的对象	subsref	重载方法
loadobj	用户对象载入函数的扩展	substruct	创建结构参数
methods	显示方法名称	superiorto	建立父类关系

Java类与对象

命令	说明	命令	说明
cell	将Java数组对象转换为单元数组	isa	判断对象是否为给定类
class	返回Java对象的类	isjava	判断是否为Java对象
clear	清除Java 包输入列表	javaArray	构造Java数组
depfun	使用M文件列出Java类	javaMethod	调用Java方法
exist	判断是否为Java 类	javaObject	构造Java对象
fieldnames	列出属于对象的公共类	methods	显示属于类的方法
import	在当前Java输入列表中添加包或类	methodsview	显示类实现的方法信息
inmem	Java类的列表名调入内存	which	显示方法的包和类名称

7. 错误处理

命令	说明	命令	说明
catch	检测语句	lastwarn	返回上次警告信息
error	显示错误信息	try	检测语句
ferror	查询文件输入输出的错误	warning	显示警告信息
lasterr	返回上次错误信息		

8. MEX编程

命令	说明	命令	说明
dbmex	进入MEX文件调试	mex	由C或Fortran源代码编译MEX文件
inmem	返回当前调入的MEX文件名	mexext	返回MEX文件名扩展

A.4　文件输入输出

A.4.1　文件名结构

命令	说明	命令	说明
fileparts	返回路径文件名等信息	tempdir	返回系统临时目录名
filesep	返回路径分割符（\）	tempname	返回临时文件名字符串
fullfile	生成全路径文件名		

A.4.2　打开、调入及保存文件

命令	说明	命令	说明
importdata	调入各种格式的数据文件	open	使用某种编辑器或程序打开文件
load	从 MAT 或 ASCII 文件中调入数据	save	将数据保存在MAT 或 ASCII 文件中

A.4.3　低层文件输入输出

命令	说明	命令	说明
fclose	关闭已打开文件	fread	读取二进制数据
feof	测试文件结尾	frewind	重定位打开的文件
ferror	查询文件输入输出错误	fscanf	按指定格式读文件
fgetl	从文件中读入一行，不保留换行符	fseek	设置文件定位指示
fgets	从文件中读入一行，保留换行符	ftell	取得文件定位指示
fopen	打开文件	fwrite	按二进制写文件
fprintf	按指定格式写文件		

A.4.4 文本文件

命令	说明	命令	说明
csvread	读取逗号分割的文本数据	dlmwrite	写入自定义分割的文本数据
csvwrite	写入逗号分割的文本数据	textread	读取指定每一数据格式的文本数据
dlmread	读取自定义分割的文本数据		

A.4.5 电子数据表

1. Microsoft Excel函数

命令	说明	命令	说明
xlsfinfo	判断文件是否为Microsoft Excel数据	xlsread	读取Microsoft Excel 数据

2. Lotus123函数

命令	说明	命令	说明
wk1read	读取Lotus123 WK1文件	wk1write	写入Lotus123 WK1文件

A.4.6 科学数据

1. 通用数据格式(CDF)

命令	说明	命令	说明
cdfinfo	返回CDF文件信息	cdfread	读取 CDF 文件

2. 自适应数据转换系统

命令	说明	命令	说明
fitsinfo	返回FITS文件信息	fitsread	读取FITS 文件

3. 分层数据格式(HDF)

命令	说明	命令	说明
hdf	HDF 文件接口	hdfread	读取HDF 文件
hdfinfo	返回HDF 或 HDF-EOS文件信息		

A.4.7 音频和视频

1. 通用函数

命令	说明	命令	说明
audioplayer	创建音频播放器对象	mu2lin	μ法则编码信号转换为线性音频信号
audiorecorder	执行实时音频捕捉	sound	将矢量转换为声音
beep	产生声音	soundsc	播送归一化声音
lin2mu	线性音频信号转换为μ法则编码信号		

2. SPARC 工作站特殊声音函数

命令	说明	命令	说明
auread	读 NeXT/SUN (.au) 声音文件	auwrite	写 NeXT/SUN (.au) 声音文件

3. Microsoft WAVE 声音函数

命令	说明	命令	说明
wavplay	播放声音	wavrecord	记录声音
wavread	读 Microsoft WAVE (.wav) 声音文件	wavwrite	写 Microsoft WAVE (.wav) 声音文件

4. AVI 函数

命令	说明	命令	说明
addframe	在AVI文件中加入一帧	aviread	读AVI 文件
avifile	创建新的AVI 文件	close	关闭AVI 文件
aviinfo	返回AVI文件信息	movie2avi	用MATLAB电影创建AVI 电影

A.4.8 图像

命令	说明	命令	说明
imfinfo	返回图像文件信息	imwrite	写图像文件
imread	读取图像文件		

A.5 图形功能

A.5.1 基本图形功能

命令	说明	命令	说明
box	封闭坐标轴	plot3	三维绘图
errorbar	误差条图	plotyy	两个Y轴的二维绘图
hold	保留当前图形	semilogx	单对数坐标绘图（X）
loglog	双对数坐标绘图	semilogy	单对数坐标绘图（Y）
polar	极坐标绘图	subplot	绘图子图
plot	二维绘图		

A.5.2 图形注释

命令	说明	命令	说明
clabel	等高线绘图加注标识	title	图形标题
datatick	图标的数据格式	xlabel	X轴标识
gtext	用鼠标在图形中定位输入文本	ylabel	Y轴标识
legend	图例	zlabel	Z轴标识
texlabel	字符标识		

A.5.3 特殊绘图

1. 面积、柱和饼图

命令	说明	命令	说明
area	绘制区域图	bar3h	水平三维柱状图
bar	垂直柱状图	pareto	Pareto 图
barh	水平柱状图	pie	绘制饼图
bar3	垂直三维柱状图	pie3	绘制三维饼图

2. 等高线绘图

命令	说明	命令	说明
contour	等高线绘图	ezcontour	等高线函数绘图
contourc	等高线计算	ezcontourf	填充等高线函数绘图
contourf	填充等高线绘图		

3. 方向和矢量绘图

命令	说明	命令	说明
comet	彗星图	feather	羽毛图
comet3	三维彗星图	quiver	矢量图
compass	罗盘图	quiver3	三维矢量图

4. 离散数据绘图

命令	说明	命令	说明
stem	绘制离散序列图	stairs	阶梯图
stem3	绘制离散表面图		

5. 函数绘图

命令	说明	命令	说明
ezcontour	等高线函数绘图	ezplot3	三维曲线函数绘图
ezcontourf	填充等高线函数绘图	ezpolar	极坐标函数绘图
ezmesh	三维网格函数绘图	ezsurf	三维彩色表面函数绘图
ezmeshc	三维网格/等高线函数绘图	ezsurfc	表面/等高线函数绘图
ezplot	简易函数绘图	fplot	函数绘图

6. 直方图

命令	说明	命令	说明
hist	绘制直方图	rose	扇形直方图
histc	直方图记数		

7. 多边形和表面绘图

命令	说明	命令	说明
convhull	凸面图	pcolor	伪彩色绘图
cylinder	圆柱体	polyarea	多边形面
delaunay	Delaunay 三角分解	ribbon	带状图
dsearch	最近点搜索	slice	立体切片图
ellipsoid	椭圆体	sphere	球体
fill	填充二维多边形	tsearch	最接近的Delaunay 三角搜索
Fill3	填充三维多边形	voronoi	Voronoi 图表
inpolygon	判断点是否在多边形内	waterfall	三维瀑布图

8. 分散点绘图

命令	说明	命令	说明
plotmatrix	分散点绘图矩阵	scatter3	三维分散点绘图
scatter	分散点绘图		

A.5.4　二进制图像

命令	说明	命令	说明
frame2im	将电影画面转为图像	im2frame	将图像转为电影画面
image	显示图像对象	imread	从图形文件读取图像
imagesc	归一数据显示图像对象	imwrite	向图形文件写入图像
imfinfo	图形文件信息	ind2rgb	将索引图像转为RGB 图像

A.5.5　打印

命令	说明	命令	说明
orient	打印纸张方向	printopt	设置本地打印机
pagesetupdlg	纸张设置对话框	printpreview	打印预览
print	打印图形或图形文件	saveas	按指定输出格式保存图形
printdlg	打印对话框		

A.5.6　图形句柄

1. 查找和识别图形对象

命令	说明	命令	说明
allchild	查找指定对象的全部子类	gcbf	返回包括调用对象的图形句柄
copyobj	复制图形对象及其子类	gco	返回当前对象的句柄
delete	删除文件或图形对象	get	获得对象属性
findall	查找全部图形对象（包括隐藏句柄）	ishandle	当值是有效的对象句柄时为真
findobj	查找特定属性值的对象	rotate	按给定点和方向旋转对象
gca	获得当前轴句柄	set	设置对象属性
gcbo	返回当前运行的调用对象		

2. 对象创建函数

命令	说明	命令	说明
axes	创建坐标轴对象	patch	创建块状绘图对象
figure	创建图形窗口	rectangle	创建矩形对象
image	创建图像矩阵	surface	创建曲面
light	创建光照对象	text	创建文本对象
line	创建曲线	uicontextmenu	创建关系菜单

3. 图形窗口

命令	说明	命令	说明
capture	当前图形的屏幕捕捉	drawnow	完成未完的绘图
clc	清除图形窗口	gcf	获得当前图形句柄
clf	清除图形	newplot	预测NextPlot属性的M文件
close	关闭指定窗口	refresh	刷新图形
closereq	关闭请求函数	saveas	按指定输出格式保存图形

4. 坐标轴操作

命令	说明	命令	说明
axis	坐标轴的刻度和外观	gca	获得当前坐标轴的句柄
cla	清除坐标轴	grid	网格线

A.6 三维可视化

A.6.1 表面和网格绘制

1. 创建表面和网格

命令	说明	命令	说明
hidden	设置消隐方式	surfc	曲面/等高线混合绘图
meshc	网格/等高线混合绘图	surfl	带光照的三维曲面图
mesh	三维网格曲面	tetramesh	四面体网格绘图
peaks	双峰函数绘图	trimesh	三角形网格绘图
surf	三维曲面图	triplot	二维三角形绘图
surface	创建曲面	trisurf	三角形曲面绘图

2. 区域生成

命令	说明	命令	说明
griddata	数据网格和曲面拟合	meshgrid	产生三维绘图网格

3. 色彩操作

命令	说明	命令	说明
brighten	色彩图的亮度控制	rgb2hsv	将RGB转换为HSV
caxis	伪彩色轴刻度	rgbplot	绘制色彩图
colorbar	显示彩色条	shading	表面色彩渲染
colordef	设置缺省颜色	spinmap	色彩图周期变化
colormap	色彩图	surfnorm	曲面法线
graymon	灰度显示器的图形设置	whitebg	设置绘图的背景色
hsv2rgb	将HSV转换为RGB		

4. 色彩图

命令	说明	命令	说明
autumn	连续变化的黄红色调图	hsv	连续变化的饱和色图
bone	连续变化的兰色调灰度图	jet	连续变化的变异饱和色图
contrast	增强对比度的灰度色彩图	lines	交错变化的线图色图
cool	连续变化的冷色调图	prism	交错变化的光谱色图
copper	连续变化的青铜色调图	spring	连续变化的黄粉色调图
flag	交错变化的红、白、蓝、黑四色图	summer	连续变化的黄绿色调图
gray	连续变化的灰度图	winter	连续变化的兰绿色调图
hot	连续变化的暖色调图		

A.6.2 视角控制

1. 控制照相机视角

命令	说明	命令	说明
camdolly	移动照相机的位置和目标位置	camtarget	照相机目标
camlookat	指定对象的视角	camup	设置和捕获照相机向上分量
camorbit	照相机目标的轨道	camva	照相机视角
campan	旋转照相机	camzoom	照相机变焦
campos	照相机位置	view	三维图形视图
camproj	设置和捕获摄影类型	viewmtx	生成视图转换矩阵
camroll	旋转照相机视轴		

2. 设置外观比例和轴范围

命令	说明	命令	说明
daspect	设置外观比例	ylim	设置和获得当前Y轴范围
pbaspect	设置和获得绘图区外观比例	zlim	设置和获得当前Z轴范围
xlim	设置和获得当前X轴范围		

3. 对象操作

命令	说明	命令	说明
reset	坐标轴或图形复位	selectmoveresize	对象的选择、移动和调整大小
rotate3d	三维图形旋转视图	zoom	图形缩放

4. 选择指定区域

命令	说明	命令	说明
dragrect	鼠标选定区域	rbbox	选定区域

A.6.3 光照

命令	说明	命令	说明
camlight	创建照相机光源	lighting	光照模式
light	创建光源	material	材料反射系数
lightangle	在球坐标中定位光照		

A.6.4 透视

命令	说明	命令	说明
alpha	当前坐标轴内对象的透视属性	alim	设置或查询坐标轴的α图范围
alphamap	指定图形的α图		

A.6.5 体积可视化

命令	说明	命令	说明
coneplot	在三维空间绘制速度矢量	slice	绘制切片图
contourslice	在体视平面绘制等高线	smooth3	平滑三维数据
curl	计算速度矢量的曲率和角度	stream2	计算二维流线数据
divergence	计算矢量场的发散角	stream3	计算三维流线数据
flow	产生三维数组	streamline	绘制流线图
interpstreamspeed	矢量场幅值产生流线	streamparticles	绘制流线点图
isocaps	计算等曲面	streamribbon	绘制流线带图
isocolors	计算等曲面的色彩	streamslice	绘制切片流线图
isonormals	计算等曲面的法线	streamtube	绘制流管图
isosurface	提取等曲面数据	surf2patch	表面数据转为片状数据
reducepatch	减少块状面的数量	subvolume	提取数据子集
reducevolume	减少体数据的元素	volumebounds	返回坐标和色彩范围
shrinkfaces	减少块状面的尺寸		

A.7　创建图形用户界面

A.7.1　预定义对话框

命令	说明	命令	说明
dialog	创建对话框	questdlg	创建提问对话框
errordlg	创建错误对话框	uigetfile	显示回调读入文件名对话框
helpdlg	显示帮助对话框	uiputfile	显示回调写入文件名对话框
inputdlg	创建输入对话框	uisetcolor	设置色彩对话框
listdlg	创建列表对话框	uisetfont	设置字体对话框
msgbox	创建信息对话框	waitbar	显示等待条
pagedlg	创建页面布局对话框	warndlg	创建警告对话框
printdlg	创建打印对话框		

A.7.2　配置图形用户界面

命令	说明	命令	说明
guidata	存储应用数据	movegui	移动GUI界面
guihandles	创建句柄结构	openfig	打开GUI 界面

A.7.3　用户界面开发

命令	说明	命令	说明
guide	打开GUI编辑器	inspect	显示属性

1. 使用应用数据

命令	说明	命令	说明
getappdata	获得应用数据	rmappdata	移动应用数据
isappdata	如果应用数据存在为真	setappdata	指定应用数据

2. 交互式输入

命令	说明	命令	说明
ginput	使用鼠标输入	waitforbuttonpress	等待键盘或鼠标事件

A.7.4　用户界面对象

命令	说明	命令	说明
menu	创建菜单	uicontrol	创建用户界面控制
uicontextmenu	创建关系菜单	uimenu	创建用户界面菜单

A.7.5　查找和识别对象

命令	说明	命令	说明
findall	查找图形对象	gcbf	返回包括调用对象的图形句柄
findfigs	显示离屏可视图形窗口	gcbo	返回执行调用的对象句柄

A.7.6　GUI设置函数

命令	说明	命令	说明
selectmoveresize	选择、移动和调整图形对象	textwrap	返回重叠字符矩阵

A.7.7　程序执行控制

命令	说明	命令	说明
uiresume	恢复中断程序运行	uiwait	暂停程序运行，由uiresume恢复

附录 B　符号数学工具箱命令和函数索引

微积分

命令	说明	命令	说明
diff	微分	limit	表达式的极限
int	积分	symsum	数列求和
jacobian	Jacobian 矩阵	taylor	Taylor 级数展开

线性代数

命令	说明	命令	说明
colspace	列空间	null	零矩阵
det	行列式	poly	定义多项式
diag	创建对角阵	rank	矩阵的秩
eig	特征值和特征矢量	rref	缩减行的梯形形式
expm	矩阵的幂	svd	奇异值分解
inv	逆矩阵	tril	下三角矩阵
jordan	Jordan 经典形式	triu	上三角矩阵

化简

命令	说明	命令	说明
collect	同类项合并	numden	分式通分
expand	符号表达式的展开	simple	寻找表达式的最简型
factor	因式分解	simplify	符号表达式的化简
horner	嵌套形式重写	subexpr	符号替换

求解方程

命令	说明	命令	说明
compose	多重函数	finverse	求反函数
dsolve	解微分方程	solve	解代数方程

变量精度

命令	说明	命令	说明
digits	设置变量精度	vpa	变量精度

算术运算

命令	说明	命令	说明
+	加法	\	左除
-	减法	.\	数组左除
*	乘法	^	矩阵或标量的幂
.*	数组乘法	.^	数组的幂
/	右除	'	转置，共轭转置
./	数组右除	.'	非共轭转置

特殊函数

命令	说明	命令	说明
cosint	余弦积分，Ci(x).	sinint	正弦积分，Si(x).
hypergeom	通用超几何函数	zeta	Riemann ζ 函数
lambertw	$\lambda(x)e^{\lambda(x)}=x$ 的解		

调用Maple

命令	说明	命令	说明
maple	调用Maple内核	mhelp	Maple帮助
mapleinit	初始化Maple	mfunlist	Mfun的函数列表
mfun	函数的数值解	procread	装入Maple程序

函数绘图

命令	说明	命令	说明
ezcontour	等高线函数绘图	ezpolar	极坐标函数绘图
ezcontourf	填充等高线函数绘图	ezsurf	三维彩色表面函数绘图
ezmesh	三维网格函数绘图	ezsurfc	表面/等高线函数绘图
ezmeshc	三维网格/等高线函数绘图	funtool	（图形化）函数计算器
ezplot	简易函数绘图	rsums	Riemann求和
ezplot3	三维曲线函数绘图	taylortool	（图形化）Taylor 级数计算器

数据变换

命令	说明	命令	说明
char	将符号对象转换为字符串	poly2sym	将系数矢量转换为符号多项式
double	将符号矩阵转换为双精度数组	sym2poly	将符号多项式转换为系数矢量

基本操作

命令	说明	命令	说明
ccode	用C代码表示符号表达式	latex	用LaTeXC代码表示符号表达式
conj	复数共轭	pretty	类似书写习惯显示符号表达式
findsym	查找符号变量	real	复数实部
fortran	用Fortran代码表示符号表达式	sym	创建符号对象
imag	复数虚部	syms	创建多个符号对象

积分变换

命令	说明	命令	说明
fourier	傅里叶变换	iztrans	Z 逆变换
ifourier	傅里叶逆变换	laplace	Laplace 变换
ilaplace	Laplace 逆变换	ztrans	Z 变换

附录 C　TeX 字符索引

字符串	符号	字符串	符号	字符串	符号
\alpha	α	\upsilon	υ	\sim	~
\beta	β	\phi	ϕ	\leq	≤
\gamma	γ	\chi	χ	\infty	∞
\delta	δ	\psi	ψ	\clubsuit	♣
\epsilon	ε	\omega	ω	\diamondsuit	♦
\zeta	ζ	\Gamma	Γ	\heartsuit	♥
\eta	η	\Delta	Δ	\spadesuit	♠
\theta	θ	\Theta	Θ	\leftrightarrow	↔
\vartheta	ϑ	\Lambda	Λ	\leftarrow	←
\iota	ι	\Xi	Ξ	\uparrow	↑
\kappa	κ	\Pi	Π	\rightarrow	→
\lambda	λ	\Sigma	Σ	\downarrow	↓
\mu	μ	\Upsilon	Υ	\circ	∘
\nu	ν	\Phi	Φ	\pm	±
\xi	ξ	\Psi	Ψ	\geq	≥
\pi	π	\Omega	Ω	\propto	∝
\rho	ρ	\forall	∀	\partial	∂
\sigma	σ	\exists	∃	\bullet	•
\varsigma	ς	\ni	∋	\div	÷
\tau	τ	\cong	≅	\neq	≠
\equiv	≡	\approx	≈	\aleph	ℵ
\Im	ℑ	\Re	ℜ	\wp	℘
\otimes	⊗	\oplus	⊕	\oslash	∅
\cap	∩	\cup	∪	\supseteq	⊇
\supset	⊃	\subseteq	⊆	\subset	⊂
\int	∫	\in	∈	\o	o
\rfloor	⌋	\lceil	⌈	\nabla	∇
\lfloor	⌊	\cdot	·	\ldots	…
\perp	⊥	\neg	¬	\prime	′
\wedge	∧	\times	×	\0	∅
\rceil	⌉	\surd	√	\mid	∣
\vee	∨	\varpi	ϖ	\copyright	©
\langle	⟨	\rangle	⟩		

责任编辑　何新权

封面设计　王凌波

责任校对　肖　宇

责任印制　宋克学